

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

**WEB-ИНТЕРФЕЙС ДЛЯ СИСТЕМЫ ЦИКЛОВОГО
ПЛАНИРОВАНИЯ НА ОСНОВЕ ORACLE PRIMAVERA P6**

Выпускная квалификационная работа
по направлению подготовки 09.03.03 Прикладная информатика
профилю подготовки «Прикладная информатика в экономике»

Идентификационный номер ВКР: 154

Екатеринбург 2019

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ
Заведующий кафедрой ИС
_____ И. А. Сулова
« ____ » _____ 2019 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
WEB-ИНТЕРФЕЙС ДЛЯ СИСТЕМЫ ЦИКЛОВОГО
ПЛАНИРОВАНИЯ НА ОСНОВЕ ORACLE PRIMAVERA P6**

Исполнитель:

обучающийся группы ИЭ-402п

А. С. Куимова

Руководитель:

ст. преподаватель каф. ИС

С. В. Ченушкина

Нормоконтролер:

ст. преподаватель каф. ИС

Н. В. Хохлова

Екатеринбург 2019

АННОТАЦИЯ

Выпускная квалификационная работа состоит из web-интерфейса для системы циклового планирования публичного акционерного общества «Уральский завод тяжелого машиностроения» (ПАО «Уралмашзавод») и пояснительной записки на 66 страницах, содержит 19 рисунков, 8 таблиц, 37 источников литературы, а также 2 приложения на 15 страницах.

Ключевые слова: ЦИКЛОВОЕ ПЛАНИРОВАНИЕ, WEB-ИНТЕРФЕЙС, АВТОМАТИЗАЦИЯ ПРОЦЕССОВ, ПЛАНИРОВАНИЕ ПРОИЗВОДСТВА.

Куимова А. С., Web-интерфейс для системы циклового планирования на основе Oracle Primavera P6: выпускная квалификационная работа / А. С. Куимова; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2019. — 66 с.

Цель работы — разработать web-интерфейс для работы с системой циклового планирования на основе Primavera P6.

Для достижения цели был проведен анализ требований к разрабатываемому продукту, исследована предметная область, проведен анализ бизнес-процессов и процессов циклового планирования предприятия, изучены возможности и технологические решения Primavera P6; спроектирована возможность интеграции и проектные решения.

Разработанный web-интерфейс используется сотрудниками отдела планирования производства ПАО «Уралмашзавод».

СОДЕРЖАНИЕ

Введение.....	5
1 Проблемы и тенденции развития информационных систем в области циклового планирования	8
1.1 Анализ источников по теме работы	8
1.2 Сущность и виды процессов в цикловом планировании	10
1.3 Обоснование необходимости и цели использования программного обеспечения для циклового планирования.....	13
2 Анализ текущего состояния информационной системы и проектирование возможности интеграции.....	16
2.1 Характеристика деятельности предприятия и отдела планирования производства	16
2.2 Описание возможностей и технологических решений Primavera Р6	18
2.3 Описание проекта системы	21
2.4 SWOT-анализ.....	22
2.5 Рекомендация по совершенствованию информационной системы.....	24
2.6 Обоснование проектных решений.....	25
3 Описание web-интерфейса системы.....	28
3.1 Общее описание разработанного программного обеспечения	28
3.2 Технологии реализации	28
3.4 Описание программных модулей	31
3.5 Расчет экономической эффективности	38
3.5.1 Описание продукции	38
3.5.2 Оценка затрат на разработку программного продукта	39
3.5.3 Расчет годового фонда времени работы на персональном компьютере	42
3.5.4 Предполагаемая цена программного продукта с учетом нормы прибыли.....	45

3.5.5 Расчет экономической эффективности программы	45
Заключение	47
Список использованных источников	48
Приложение А	52
Приложение Б	54

ВВЕДЕНИЕ

На сегодняшний день планирование экономической деятельности всех предприятий является важной предпосылкой эффективного производства, независимо от сферы деятельности и масштабов организации.

Назначение планирования как функции управления состоит в стремлении заблаговременно учесть по возможности все внутренние и внешние факторы, обеспечивающие благоприятные условия для нормального функционирования и развития предприятия. Оно предусматривает разработку комплекса мероприятий, определяющих последовательность достижения конкретных целей с учетом возможностей наиболее эффективного использования ресурсов каждым производственным подразделением и в целом предприятием.

Планирование призвано обеспечить связь между отдельными структурными подразделениями предприятия, включающими всю технологическую цепочку: научные исследования и разработки, производство и сбыт. Эта деятельность опирается на выявление и прогнозирование потребительского спроса, анализ и оценку имеющихся ресурсов и перспектив развития хозяйственной конъюнктуры.

В современных условиях хозяйствования машиностроительное предприятие следует рассматривать как производственную систему, представляющую собой объективно существующий комплекс материальных объектов, коллективов людей, производственных, научно-технических и информационных процессов, имеющих целью выпуск промышленной продукции определенного назначения и качества. При этом процесс управления производством представляется как иерархическая интегрированная система, реализующая свои функции и задачи управления на уровне предприятия, производства, цеха, участка, бригады конкретным органом управления.

Целью планирования является не только создание условий для согласованной работы всех производственных подразделений по выпуску заказан-

ной потребителями продукции, но и соблюдение установленных значений технико-экономических показателей, таких как затраты, фонд зарплаты, себестоимость, производительность труда и т.д.

Важная роль принадлежит планированию, в процессе которого следует разрабатывать комплекс взаимосвязанных планов по всем производственным структурам предприятия, вплоть до рабочего места, и периодам времени, не только согласованными с планами отгрузки готовой продукции по договорам потребителей, но и допускающими сменно-суточный, а при необходимости и часовой режим регулирования.

На сегодняшний день на предприятии ПАО «Уралмашзавод» реализован комплекс программных средств, при помощи которых реализуется цикловое планирование производства. Средством ввода информации в базу данных и средством визуализации является Primavera P6. Обработка введенных данных и расчет циклового плана производства осуществляется в Microsoft SQL Server.

Primavera P6 [14] является системой управления проектами, разработанной для больших предприятий. Primavera P6 предназначена для автоматизации процессов управления проектами в соответствии с требованиями Project Management Institute (PMI), International Project Management Association (IPMA) и стандартами International Organization for Standardization (ISO). Все программные модули Primavera P6 обеспечивают хранение и обработку данных по всем проектам компании в едином хранилище данных, построенном на базе Microsoft SQL Server.

В Primavera P6 отсутствует возможность расширить функционал системы, необходимый для работы циклового планирования. В настоящее время запуском расчета циклового плана, обновлением шаблонов производства и другими необходимыми функциями занимается администратор базы данных, что очень неудобно и для самого администратора, и для отдела планирования.

Чтобы сделать работу отдела планирования производства более эффективной и снизить нагрузку на администратора базы данных, решено было разработать web-интерфейс для системы циклового планирования ПАО «Уралмашзавод», которое позволит сотрудникам работать быстрее, эффективнее и не допускать логических и структурных ошибок.

Актуальность данной работы обусловлена тем, что возникла потребность в разработке программного обеспечения, которое позволило бы улучшить существующую систему циклового планирования ПАО «Уралмашзавод».

Объектом выпускной квалификационной работы является управление цикловым планированием производства ПАО «Уралмашзавод».

Предметом выпускной квалификационной работы является расширение функционала системы циклового планирования Primavera P6.

Цель работы — разработать web-интерфейс для работы с системой циклового планирования Primavera P6.

В соответствии с поставленной целью в работы определены следующие задачи:

1. Исследовать предметную область, провести анализ бизнес-процессов и процессов циклового планирования предприятия.
2. Проанализировать литературные и интернет источники посвященные аналогичным программным продуктам.
3. Изучить возможности и технологические решения Primavera P6.
4. Спроектировать возможность интеграции и проектные решения.
5. Реализовать и внедрить разработанный web-интерфейс для работы с реальными данными предприятия.

1 ПРОБЛЕМЫ И ТЕНДЕНЦИИ РАЗВИТИЯ ИНФОРМАЦИОННЫХ СИСТЕМ В ОБЛАСТИ ЦИКЛОВОГО ПЛАНИРОВАНИЯ

1.1 Анализ источников по теме работы

В современном мире невозможно представить себе предприятие, которое не использует информационные системы для автоматизации бизнес процессов, подробно данная тема рассматривается в статьях Г. Б. Евгеньева, С. С. Крюкова, Б. В. Кузьмина и А. Г. Стисес «Интегрированная система автоматизации проектирования технологических процессов и оперативного управления производством» [7], В. В. Буторова и С. П. Ереско «Автоматизация параметрического моделирования в машиностроительном производстве» [1], А. А. Брыксиной и Н. Е. Проскурякова «Анализ технологии и разработка методов повышения энергоэффективности производства» [2] и в статье А. Н. Трусова «Модель экономической эффективности автоматизации конструкторско-технологической подготовки машино-строительного производства» [10].

Автоматизация процесса планирования на производстве подробно рассмотрена О. В. Колобиной в статье «Принципы стратегического планирования на предприятии» [20]. В статье говорится о том, что автоматизация имеет важнейшее значения для всего процесса производства, в частности, для процесса планирования.

Основой успеха предприятия в настоящее время является тщательно проработанный, просчитанный и обоснованный план производства. Руководителям, менеджерам и планировщикам, подготавливающим планы производства, а также отвечающим за их точность и своевременность, необходимо программное обеспечение, способное оказать помощь в повышении продуктивности их предприятий.

Также в статье говорится о том, что основным плюсом автоматизации процесса планирования является возможность составления нескольких планов производства с различными параметрами, сравнение созданных планов и выбор лучшего. До автоматизации план производства составляется «вручную», на большом производстве это достаточно долгий и трудоемкий процесс, который может занимать не один день. Внести какие-либо изменения при таком способе планирования очень сложно. Составить несколько различных планов и выбрать лучший — практически невозможно.

Подобные выводы также можно сделать, проанализировав статьи С. А. Овшинова «Оперативно-календарное планирование в серийном производстве» [11], О.В. Косенко «Распределение последовательности подзадач при календарном планировании производства» [25], Д. М. Гамалей и И. В. Филько «Планирование производства продукции в условиях конкурентной среды» [17].

В статье А. Шуленина на тему «Microsoft SQL Server и активный Internet» [32] подробно рассматриваются различные способы работы с базой данных через Интернет и большое разнообразие средств для реализации такого взаимодействия. А. Шуленин рассказывает о том, что в современном мире очень удобно иметь возможность работать с базой данных просто открыв нужный сайт в браузере, вместо загрузки, как правило, занимающих большое количество памяти стандартных средств управления базами данных. Также это частично решает проблему исключения ошибок со стороны пользователя при вводе и редактировании данных. Дополнительно к плюсам такого решения можно отнести мобильность рабочего места пользователя, поскольку для доступа к информации из базы данных необходим только выход в Интернет.

Статья А. Проскониной «Web-приложения и данные: проблемы абстракции и масштабируемости» [34] интересна тем, что в ней достаточно подробно описана архитектура Модель-Вид-Контроллер (Model-View-Controller, MVC), которая наиболее часто используется для web-интерфейсов.

Информацию о современном построении web-интерфейса можно также найти в статьях И. А. Ботыгина и К. А. Каликина «Исследование методов увеличения производительности web-приложений» [9], Е. В. Прокопенко и Т. В. Сарапуловой «Разработка web-приложений для поддержки стратегического управления» [22], а также статьи Э. Р. Набеевой и Р. Р. Кагирова «Технологии создания пользовательских web-приложений» [28].

Для выбора средств разработки был рассмотрен рейтинг лучших языков программирования по версии Techrocks [21], а также статьи С. О. Бондаренко «Современные интерактивные web-приложения — построение пользовательского интерфейса с React» [27], А. А. Карышева и В. Р. Афанасьева «Разработка web-сервиса для автоматизированной генерации документов на основе docx-шаблонов» [23], статья И. С. Скороходова и А. Н. Тихомировой «Исследование и сравнение современных реализаций flux-архитектур разработки web-приложений» [8].

Для разработки информационной системы, было необходимо изучить государственные стандарты разработки программного обеспечения:

- ГОСТ 19.102-77 «Стадии разработки» [5], в данном документе описывается весь жизненный цикл создания системы, этапы разработки;
- ГОСТ 19.105-78 «Общие требования к программным документам». ГОСТ содержит полный перечень документации, которая сопровождает законченный программный продукт [6];
- ГОСТ 19.101-77 «Виды программ и программных документов» [4], данный ГОСТ это единая система программной документации, описывающая виды программ и программных документов для программного обеспечения.

1.2 Сущность и виды процессов в цикловом планировании

Единичное производство характеризуется значительным количеством заказов на разную продукцию, но при этом могут преобладать отдельные ти-

пы работ [29]. Из этого следует, что сущность циклового планирования производства заключается в том, чтобы объединить заказы так, чтобы обеспечить оптимально сочетание сроков изготовления продукции и загрузки оборудования.

Для реализации такого подхода составляют предварительный расчет длительности производственного цикла. Результатом предварительного расчета являются примерные сроки изготовления продукции, а также информация об нагрузках на группы оборудования.

Особенностью единичного производства является прохождение всех стадий производства каждым изделием:

- научно-исследовательская работа;
- разработка конструкторской документации;
- разработка технологической документации;
- разработка маршрутной документации;
- нормирование затрат труда;
- изготовление;
- испытание.

Календарное распределение включает в себя [31]:

- расположение изделий в плане согласно срокам, предусмотренным договорами;
- группировка изделий со схожими параметрами, для параллельного изготовления с расчетом на партийное производство общих деталей;
- составление групп оборудования, имеющих функциональную общность, анализ загрузки групп оборудования для того, чтобы оптимально загрузить оборудование и площади производства;
- максимальная концентрация выпуска определенного вида изделий в интервале времени, таким образом сокращая номенклатуру изготавливаемых изделий;
- расчет плана производства в нескольких вариантах, для выбора лучшего решения;

- проверка полученного графика производства с точки зрения равномерности выпуска;
- ручная корректировка расчетов для получения наиболее эффективного плана производства.

Особенности циклового планирования производства обусловлены большой, можно сказать, неограниченной номенклатурой.

Существует позаказный метод планирования, то есть изготовление изделий осуществляется по заказам. В работу по заказу входит не только само изготовление изделия, но и конструкторско-технологическая подготовка производства.

Главной задачей циклового планирования является одновременное производства множества различных по конструкции и технологии изделий в ограниченные сроки. При этом необходимо распределить работы равномерно по всем станкам и оборудованию, чтобы обеспечить рациональное использование трудовых и нетрудовых ресурсов производства. Равномерному распределению нагрузки способствует объединение санков в группы оборудования по схожему функционалу.

В достаточно обобщенной методологической трактовке планирование производства на машиностроительном предприятии следует рассматривать как [12]:

- четкое установление заданий с учетом обоснованной цели функционирования;
- прогнозирование и точное предвидение возможного состояния дел на управляемом объекте;
- моделирование, т.е. формализованное описание процедур и порядка выполнения плановых расчетов;
- тактику реализации поставленной цели и получаемых при этом результатов.

1.3 Обоснование необходимости и цели использования программного обеспечения для циклового планирования

Сегодня, учитывая уровень прогресса и конкуренцию на рынке, сложно представить себе предприятие, на котором не использовалось бы программное обеспечение, соответствующее отрасли.

Автоматизация промышленных предприятий является основным вектором технологического развития. Прогресс предполагает такое развитие, при котором операции, ранее выполняемые людьми, выполняет специализированное оборудование и программное обеспечение, а человек осуществляет лишь контроль и управление.

Оптимизация процессов планирования — это основа деятельности предприятия, направленная на повышение производительности труда, снижение себестоимости производимой продукции, рациональное использование ресурсов, и как результат — повышение рентабельности предприятия. В силу того, что производственное планирование требует учета множества всевозможных факторов, оно является наиболее трудоемким звеном производственного учета.

Деятельность ПАО «Уралмашзавод» невозможна без составления проработанного плана производства. Для этого необходимо внести базу:

- информациях о ресурсах, их доступности и интенсивности;
- общую информацию об изделиях (изделия представлены наборами больших модулей);
- информацию о заказах клиентов;
- параметры для расчета плана.

Так же для расчета необходима информация о полном составе изделий, от основных компонентов до мелких номенклатур с технологическими процессами. Технологический процесс представляет собой информацию о том, на каком оборудовании, в какой последовательности и сколько по времени изготавливается номенклатура.

Когда речь идет о токах сложных изделиях, как экскаваторы, важно учитывать следующие особенности:

- сложный состав. Состав таких изделий можно представить в виде дерева, в вершине которого само изделие, которое разбирается на крупные составные части — модули. Такое дерево состава может иметь несколько уровней, а количество узлов может достигать тысяч или даже десятков тысяч штук;
- длительные циклы производства (как правило, изготовление каждого изделия занимает полгода, год);
- долгие сроки поставки материалов;
- сложная техническая и конструкторская документация. Нередко случаются ситуации, когда к началу производства изделия документация готова не в полном объеме и дополняться в процессе производства.

Все вышеперечисленные особенности ставят задачи планирования по важности на самый высокий уровень. Как уже было сказано выше, каждое изделие состоит из многих сотен тысяч деталей, для каждой из которых нужно подобрать станок, время и закупить материалы.

Ранее процесс планирования осуществлялся сотрудниками «вручную», то есть, основным программным обеспечением был Microsoft Office Excel, составление одного плана длилось неделями. Сегодня на ПАО «Уралмашзавод» разработана система циклового планирования [19], которая состоит из средства ввода информации в базу данных и средства визуализации Primavera P6 и средства обработки введенных данных и расчета циклового плана производства Microsoft SQL Server.

Осуществляет ввод информации при помощи Primavera P6 специалист отдела планирования, для запуска необходимых процедур специалисту отдела планирования необходимо обратиться к администратору базы данных, который, используя Microsoft SQL Server, запускает необходимые процедуры по обработке шаблонов изделий и планов производства, с параметрами, переданными специалистом отдела планирования.

Одной из проблем такого порядка является то, что у администратора базы данных не всегда есть свободное время для запуска необходимых специалисту отдела планирования процедур, также по результатам выполнения процедур, необходимо дать специалисту отдела планирования обратную связь, на что тоже необходимо выделить время. На сегодняшний день, выполнение какой-либо процедуры по обработке информации о шаблоне изделия, учитывая выше сказанное, может занимать целый рабочий день, тогда как фактически для выполнения процедуры необходимо 15–60 минут.

Проанализировав текущие проблемы, было решено разработать web - интерфейс для системы циклового планирования ПАО «Уралмашзавод», которое позволит автоматизировать процессы запуска циклового плана, нормализации шаблонов и планов производства, обновления ресурсов в шаблонах и многие другие, которые сейчас запускаются вручную администратором базы данных.

Необходимо учитывать, что потребности организации постоянно растут, в связи с этим необходимо предусмотреть возможность расширения функционала программного продукта.

2 АНАЛИЗ ТЕКУЩЕГО СОСТОЯНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ И ПРОЕКТИРОВАНИЕ ВОЗМОЖНОСТИ ИНТЕГРАЦИИ

2.1 Характеристика деятельности предприятия и отдела планирования производства

ПАО «Уралмашзавод» («УЗТМ», Уралмаш) — машиностроительное предприятие в городе Екатеринбург, основанное в июле 1933 года, одно из крупнейших в России [16]. Производитель оборудования для металлургии, горнодобывающей промышленности, промышленности строительных материалов и энергетики.

Основные виды выпускаемой продукции предприятия:

- дробильно-размольное оборудование (дробилки и мельницы разных типоразмеров) (рисунок 1);



Рисунок 1 — Дробилка

- экскаваторы различных классов и модификаций (электромеханические, шагающие, гидравлические) (рисунок 2);



Рисунок 2 — Экскаватор

- мостовые тяжелые краны;
- металлургические прессы, оснастка и инструмент для них;
- шахтные подъемные установки.

Стратегия развития Уралмашзавода предусматривает переход предприятия на уровень компании, предлагающей горнодобывающей отрасли и металлургии комплексные решения их производственных задач на базе оборудования УЗТМ и российских производителей. На заводе реализуется программа, направленная на масштабную реконструкцию механообрабатывающего и сварочного производств, цифровизацию производственных процессов.

Эффективная организация производственных процессов, во многом зависит от деятельности отдела планирования производства.

При составлении производственных планов учитываются потребности клиентов в конкретные временные промежутки для обеспечения высокого уровня обслуживания клиентов компании. Это означает, что сотрудники отдела планирования определяют в какое время и сколько нужно произвести продукции производственным участкам и контролируют исполнение планов.

В случае невыполнения планов производства предприятие несёт потери от непроданного товара в краткосрочной перспективе и от возможной потери клиентов в долгосрочной перспективе.

Переизбыток товара также приводит к дополнительным затратам предприятия, так как происходит отвлечение денежных средств в запасах, готовая продукция занимает место. Всё это свидетельствует о необходимости чёткого планирования производства для исключения проблем дефицита и больших товарных остатков при производстве.

Основными пользователями разрабатываемого продукта будут специалисты по планированию, в их обязанности входит:

- сбор необходимой технической и экономической информации, необходимой для составления планов производства;
- подготовка исходных данных для составления планов производства;
- проектирование моделей возможного развития событий;
- подготовка проектов внесения изменений в план в связи с возникновением новых факторов, не учтенных при составлении плана производства;
- участие в технико-экономическом обосновании реализации новых проектов на производстве;
- разработка мероприятий по эффективному использованию ресурсов предприятия;
- подготовка отчетов;
- выполнение работ по формированию, ведению и хранению информации для планирования.

2.2 Описание возможностей и технологических решений

Primavera P6

На данный момент ввод информации об изделиях производства осуществляется через систему управления проектами Primavera P6.

Сначала создается шаблон изделия, например, шаблон Дробилка КМД-2200 Т6-Д (рисунок 3).

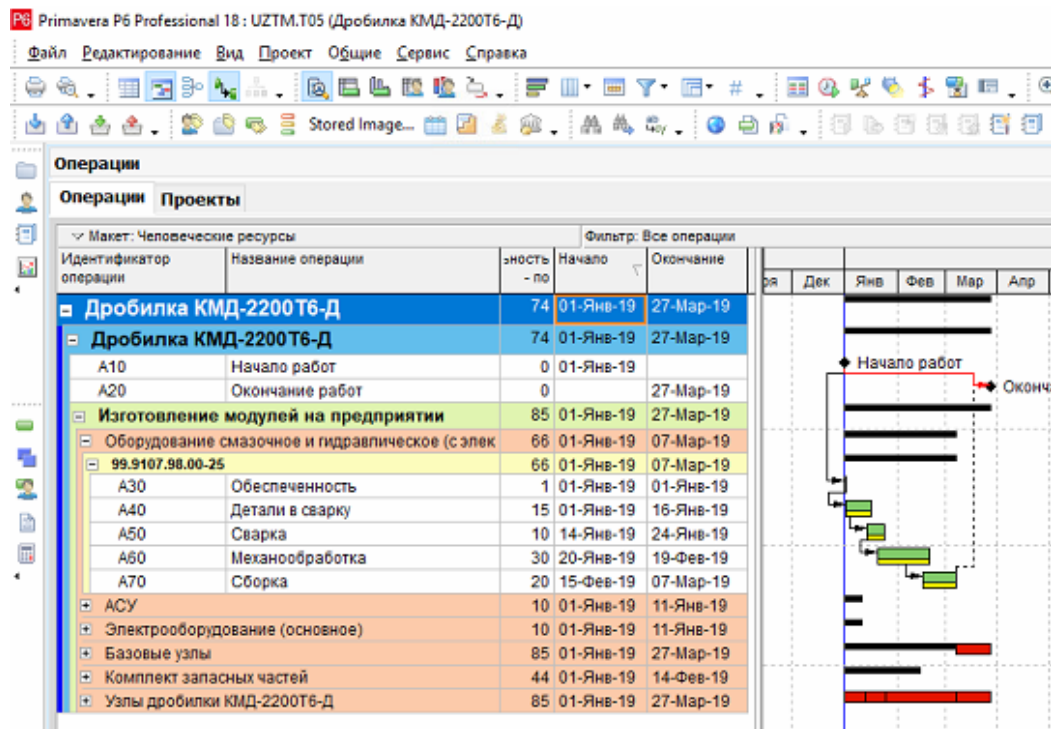


Рисунок 3 — Шаблон дробилки

Структура шаблона строго стандартизирована:

- изделие (главный модуль);
- внутри изделия создаются работы «Начало работ» и «Окончание работ» и модуль «Изготовление модулей на предприятии»;
- в «Изготовление модулей на предприятии» заносятся модули изделия, в данном случае это:
 - оборудование смазочное и гидравлическое (с электрооборудованием)
 - АСУ;
 - электрооборудование (основное);
 - базовые узлы;
 - комплект запасных частей;
 - узлы дробилки КМД-2200 Т6-Д;
- внутри каждого модуля создается номенклатура (например, 99.9107.98.00-25);

- в номенклатуре, создаются основные работы (некоторые работы могут отсутствовать, в зависимости от процесса изготовления):

- обеспеченность;
- детали в сварку;
- сварка;
- механообработка;
- сборка.

После создания шаблона, его необходимо наполнить ресурсами и расставить длительности работ, для этого системным администратором запускается специальная процедура в MS SQL Server, которая наполнит шаблон данными техпроцесса.

Из заполненных шаблонов составляются планы производства, запускаются процедура расчета так же в MS SQL Server, после чего выбирается и утверждается оптимальный план, который становится основным планом производства (рисунок 4).

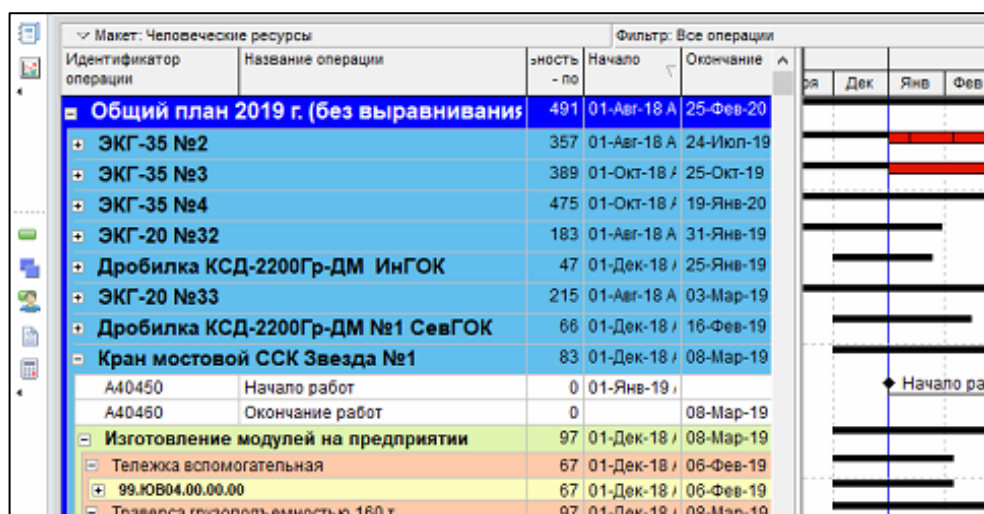


Рисунок 4 — План производства

Процесс составления и расчетов плана производства представлен в виде диаграммы (рисунок 5).

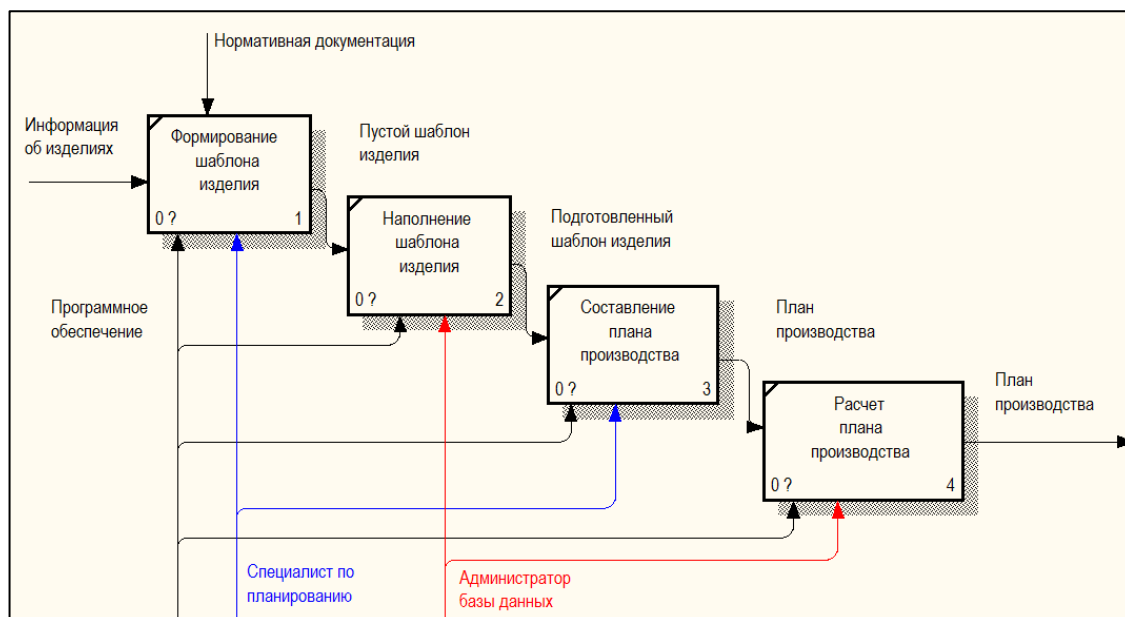


Рисунок 5 — Процесс запуска циклового планирования

Диаграмма процесса составления плана и запуска расчетов показывает, что в процессе участвуют как специалист по планированию, так и администратор базы данных.

2.3 Описание проекта системы

Процесс составления и расчета плана производства с использованием системы представлен в виде диаграммы (рисунок 6).

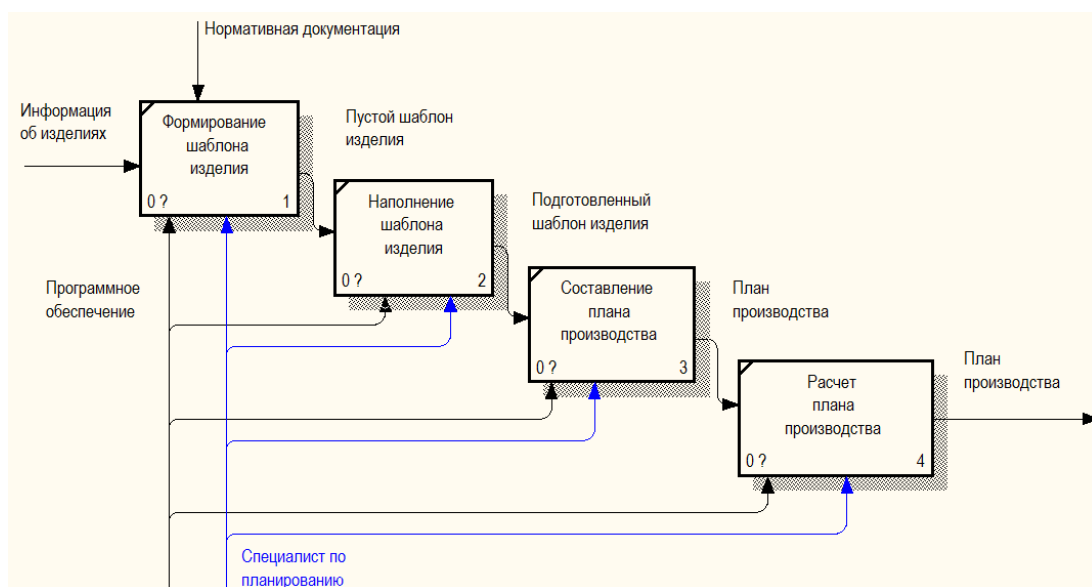


Рисунок 6 — Процесс планирования производства

Web-интерфейс должен обладать высокой скоростью обработки данных, гибкостью, надёжностью. Сложные управляемые данными web-интерфейсы по ряду причин используют систему управления базами данных (СУБД). Во-первых, с их помощью возможно переложить большинство задач хранения и управления данными на СУБД. Во-вторых, СУБД лучше человека справляются с управлением большими объемами данных. В-третьих, база данных (БД) хранят данные постоянно.

Хранение информации в базе данных также позволяет писать меньший объем кода (благодаря тому, что задачи обработки данных передаются СУБД) и рассматривать отвлеченно всю систему управления данными. Фрагмент схемы базы данных представлен на рисунке 7.

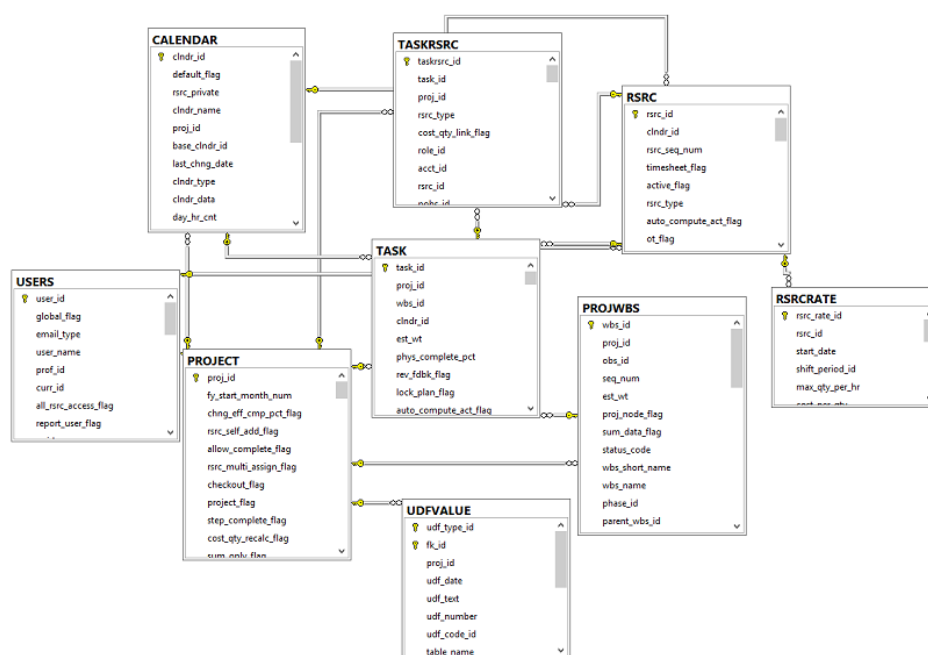


Рисунок 7 — Фрагмент базы данных

Фрагмент схемы базы данных реализован в виде ER-диаграммы.

2.4 SWOT-анализ

SWOT-анализ — это метод первичной оценки текущей ситуации основанный на рассмотрении её с четырёх сторон: Strengths — сильные стороны;

Weaknesses — слабые стороны; Opportunities — возможности; Threats — угрозы.

Основным преимуществом данного анализа является возможность определить, хватит ли у компании или программного продукта сил и ресурсов для реализации своих возможностей и реагировать на внешние угрозы без особых последствий.

SWOT-анализ невозможно составить без некоторых правил:

- необходимо выбрать максимально точную область исследования. При выборе слишком обширной области результаты анализа будут некорректными и не применимыми;
- четкое разделение элементов SWOT-анализа. Важно не путать сильные стороны и возможности;
- необходимо избегать субъективности;
- очень важно использовать мнение как можно большего количества людей. Чем больше выборка, тем точнее исследование;
- максимально конкретные и точные формулировки.

В процессе проведения анализа внутренней ситуации предприятия оцениваются ее бизнес-процессы, конкурентоспособность и ресурсы.

Проанализируем текущее состояние бизнес-процесса внесения данных и запуска расчета циклового планирования, до внедрения web-интерфейса (таблица 1).

Таблица 1 — Анализ ситуации до внедрения web-интерфейса

Внутренние сильные стороны (S)	Внутренние слабые стороны (W)
Не требуется обучение персонала. Относительно стабильная работа циклового планирования.	Запуск расчета, наполнения шаблонов, процедур нормализации и оптимизации осуществляется администратором базы данных. Неэффективное распределение рабочего времени сотрудников.
Внешние возможности (O)	Внешние угрозы (T)
Бизнес-процесс позволил организовать цикловое планирование.	Ошибки в процессе сбора, занесения, хранения информации

Далее рассмотрим ситуацию в разрезе тех же факторов после внедрения web-интерфейса (таблица 2).

Таблица 2 — Анализ ситуации после внедрения web-интерфейса

Внутренние сильные стороны (S)	Внутренние слабые стороны (W)
Удобные интерфейс. Удаленное рабочее место Возможность доработки системы под нужды предприятия. Автоматизация основных функций специалиста отдела планирования. Увеличение скорости работы персонала. Уменьшение нагрузки на администратора базы данных.	Требуется обучение персонала. Требуется постоянный, стабильный доступ в Интернет.
Внешние возможности (O)	Внешние угрозы (T)
Сокращение времени на работу с обработкой и внесением первичной информации. Использование современного программного средства.	Конкуренция на рынке ПО.

Как показывает сравнительный анализ таблиц 1 и 2 при внедрении web-интерфейса для системы циклового планирования в ПАО «Уралмашзавод» автоматизируется процесс формирования и расчета планов производства, также автоматизируются процессы запуска процедур наполнения, оптимизации и нормализации. Таким образом, SWOT-анализ выявил необходимость в разработке web-интерфейса.

2.5 Рекомендация по совершенствованию информационной системы

Проанализировав состояние информационной системы в настоящий момент времени, можно выделить следующие недостатки, которые необходимо исправить:

- процедуры наполнения, нормализации и оптимизации запускаются вручную администратором базы данных по просьбе специалиста отдела планирования, такой порядок запуска занимает большое количество времени работы специалистов;

- каждый расчет циклового плана запускается отдельно, также вручную администратором базы данных через интерфейс СУБД, что неудобно как для специалиста отдела планирования, так и для администратора базы данных;

- составление плана производства и редактирование информации об изделиях в плане специалистом отдела планирования не оптимизированно.

На основании выявленных недостатков существующей системы сформируем рекомендации для её совершенствования — разработать программный продукт, который позволит автоматизировать:

- запуск процессов оптимизации, нормализации и наполнения для шаблонов и планов;

- процедуры утверждения шаблонов и планов;

- процесс составления планов производства и внесения дополнительной информации;

- запуск расчета плана производства с учетом выбранных параметров.

Разработанный программный продукт должен значительно оптимизировать работу специалиста отдела планирования, а также снять нагрузку с администратора базы данных.

2.6 Обоснование проектных решений

Программное обеспечение — совокупность программ для реализации целей и задач информационной системы, а также нормального функционирования комплекса технических средств.

В состав программного обеспечения входят общесистемные и специальные программные продукты. К общесистемному программному обеспечению относятся комплексы программ, ориентированных на пользователей и предназначенных для типовых задач обработки информации.

Основой любого автоматизированного рабочего места является организация данных, которые подвергаются обработке. В связи с этим необходимо

обосновать выбор системы управления базами данных. Наиболее широко используемая для работы под управлением ОС Windows — СУБД Microsoft SQL Server.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основным используемым языком запросов — Transact-SQL [26].

Требования к разрабатываемому продукту предъявляются следующие:

- функциональность;
- простота использования и быстродействие;
- приятный и понятный интерфейс;
- легкость обновления;
- кроссплатформенность.

Проведем сравнительный анализ между desktop-приложениями и web-приложениями.

Под обычными приложениями будем считать приложения, установленные на локальной машине. Приложение для настольного компьютера — это программа, которая работает рабочем столе. Desktop-приложения удобны для пользователей настольных компьютеров, такие приложения обычно интегрированы с различными офисными и другими настольными приложениями.

Web-приложение — это решение, в основе которого лежит взаимодействие браузера и web-сервера. Такие приложения являются кроссплатформенными сервисами, доступными с любого современного устройства, и не привязаны к архитектуре сети: получение доступа к ним возможно с локального компьютера или со смартфона на другом конце света по удобному протоколу, например, наиболее быстрому или зашифрованному.

Web-приложение отличается от обычного web-сайта интерактивностью. Это означает, что пользователь является не пассивным потребителем, а активным участником сайта. Пользователь ищет, нажимает кнопки, заполняет формы и вообще постоянно работает клавиатурой и мышью.

Результаты сравнительного анализа представлены в виде таблицы (таблица 3).

Таблица 3 — Сравнительная таблица desktop и web

Параметр	Desktop-приложение	Web-приложение
Доступ к сети Интернет	не требуется	необходимо
Установка/обновление	Должно быть развернуто или установлено.	Единовременная настройка. Одна установка для всех пользователей. Благодаря централизованности моментально обновление.
Интерфейс взаимодействия	Стандартные интерфейсы, стандартное взаимодействие	Разнообразный интерфейс взаимодействия.
Совместимость с устройствами	Зависимость от платформы. Исключение — кроссплатформенные приложения.	Платформено-независимое.
Поиск по контенту	Нет, если только не реализовано на уровне приложения.	Да есть. Причём можно организовать свой поиск, но и воспользоваться сторонними сервисами, к примеру запрашивать данные у Google.
Совместный доступ	Если только дополнительно настроить	Изначально web-приложения (большинство) настроены на совместный доступ
Разработка	Под каждую платформу есть свои инструменты, зачастую под каждую платформу приходится писать свою версию.	Всё выполняется на сервере. Кроссплатформенно, нужен только браузер.

Из приведенной выше сравнительной таблицы и предъявляемых к программному продукту требований, мы можем сделать вывод, что web-интерфейс больше подходит для нужд ПАО «Уралмашзавод».

3 ОПИСАНИЕ WEB-ИНТЕРФЕЙСА СИСТЕМЫ

3.1 Общее описание разработанного программного обеспечения

Web-интерфейса для системы циклового планирования будет служить на ПАО «Уалмашзавод» инструментом запуска расчета циклового плана, процедур наполнения шаблонов, нормализации, оптимизации и утверждения шаблонов изделий и планов производства, инструментом формирования планов производства, что повысит эффективность работы отдела планирования и снизит нагрузку на администратора базы данных.

Функционал для работы с шаблонами производства:

- создание шаблона производства;
- редактирование шаблонов;
- утверждение шаблонов;
- наполнение шаблонов ресурсами и работами;
- нормализация и оптимизация шаблонов.

Функционал для работы с планами производства:

- формирование плана производства из утвержденных шаблонов;
- назначение производственного номера изделию в плане;
- редактирование шаблонов внутри плана;
- утверждение планов;
- обновление ресурсов в плане;
- запуск расчета циклового плана с выбором параметров.

3.2 Технологии реализации

Web-интерфейс для системы циклового планирования реализовано средствами языков Golang, Javascript и T-SQL.

При разработке внешней части web-интерфейса был использован язык Javascript с библиотекой React.js, на T-SQL реализованы возможности запуска процедур наполнения шаблонов, нормализации, оптимизации и расчета циклового плана. Golang, кроме прочих функций, осуществляет связь между внешней стороной web-интерфейса и базой данных.

Для реализации **клиентской части** использовалась библиотека React [15] была впервые выпущена компанией Facebook в 2013 году [18, 24]. За прошедшее время эта технология стала одной из самых популярных для создания web и мобильных приложений. Опрос более 50 000 разработчиков на StackOverflow [33, 34] в этом году показал, что React стала самой популярной и востребованной технологией.

React это библиотека для создания пользовательских интерфейсов. Одной из ее отличительных особенностей является возможность использовать JavaScript XML (JSX), язык программирования с близким к HyperText Markup Language (HTML) синтаксисом, который компилируется в JavaScript.

Высокой производительности приложений помогает добиться технология Virtual DOM. С React появляется возможность создавать изоморфные приложения, которые отвечают пользователю гораздо быстрее обычных. Созданные компоненты могут быть с легкостью изменены и использованы заново в новых проектах. Высокий процент переиспользования кода повышает покрываемость тестами, что, в свою очередь, приводит к более высокому уровню контроля качества.

Основные плюсы React:

- Virtual DOM повышает производительность приложений, что снижает вероятность возникновения возможных неудобств и улучшает пользовательский опыт;
- использование изоморфного подхода помогает производить загрузку страниц быстрее, тем самым позволяя пользователям чувствовать себя более комфортно во время работы с приложением.

Использование библиотеки React при проектировании web-интерфейса для системы циклового планирования, позволит создать быстрый и эффективный продукт, с которым пользователям будет приятно и удобно работать, а значит они смогут выполнять свою работу быстрее и качественнее.

Панель управления разрабатывалась с использованием Go [13] — это язык программирования, который был разработан корпорацией Google. Часто его еще называют «Golang» — сокращение от «Google language». Это компилируемый и многопоточный язык, релиз которого состоялся в ноябре 2009 года [30]. Один из основных разработчиков Go отметил, что это язык — попытка перенять лучшие стороны таких языков, как C++ и Java.

Язык Go создавался для написания различных высокоэффективных программ, однако большинство разработчиков сошлись во мнении, что он лучше всего подходит для создания web-приложений в качестве back-end.

Основные плюсы Go:

- простота. Этот язык разрабатывался в качестве замены C, а значит по производительности Go ничем не уступает, при этом дает возможность разрабатывать приложения гораздо быстрее;
- большое количество библиотек, позволяющих выполнить практически любую задачу, что необходимо при написании web-интерфейса для системы циклового планирования, которое нужно будет связать с базой на MS SQL;
- Go это компилируемый язык, который имеет строгую статическую типизацию. Одной из особенностей компилируемых языков является их высокая производительность.

Go позволит web-интерфейсу для системы циклового планирования быстро отправлять запросы на сервер и получать ответы, что необходимо для эффективной работы системы циклового планирования.

При помощи широкого выбора библиотек Go, есть возможность быстро формировать различные отчеты и сразу отправлять их пользователю, не занимая память сервера готовыми файлами и шаблонами.

3.4 Описание программных модулей

Разработанный web-интерфейс состоит из следующих вкладок:

- «Страница авторизации»;
- «Формирование шаблона»;
- «Формирование плана»;
- «Редактирование плана»;
- «Расчет плана»;
- «Отчеты»;
- «Ресурсы».

Вкладка формирование шаблона позволяет пользователям web-интерфейса редактировать шаблоны изделий, создавать новые и запускать процедуры наполнения, нормализации и оптимизации шаблонов, так же реализована возможность утверждения шаблона.

Процесс составления производственного плана из шаблонов реализован на вкладке формирование шаблона, также при добавлении шаблонов в план производства необходимо по каждому шаблону вносить дополнительную информацию, включающую в себя заводской номер изделия, дату выполнения по договору и тип изделия.

В шаблоны внутри конкретного производственного плана могут вноситься различные изменения, это возможно сделать на вкладке редактирование плана.

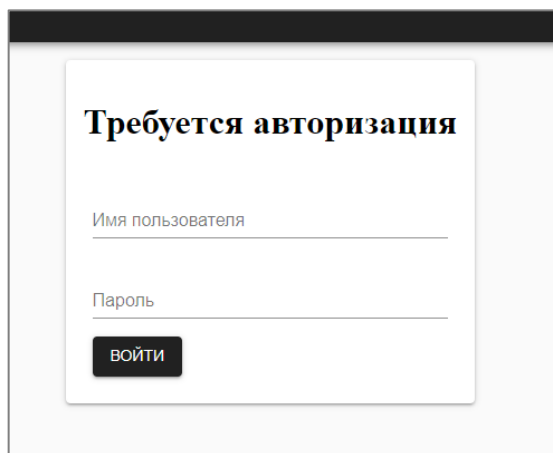
На вкладке отчеты реализована функция формирования и скачивания отчетов. Для отчетов, которые формируются более часа, отдельно реализована функция запуска с параметрами. Сформированную информацию можно скачать с этой же вкладки, для этого присутствует таблица с доступными для скачивания отчетами.

Вкладка ресурсы позволяет пользователям получить информацию о доступных ресурсах предприятия и редактировать их, изменяя группирование ресурсов, их наименование и интенсивность.

Примеры кодов клиентской и серверной части для основных элементов представлены в приложении Б.

Страница авторизации

Страница авторизации представляет собой обычную форму авторизации (рисунок 8).



Требуется авторизация

Имя пользователя

Пароль

ВОЙТИ

Рисунок 8 — Форма авторизации

Формирование шаблона

Вкладка формирование шаблона представляет собой таблицу, в которой перечислены все шаблоны, находящиеся в разработке (рисунок 9).

Формирование шаблона					Создать новый шаблон	
Код проекта	Название проекта	Степень проработки	Статус			
OMZ-CC.T01.	Фланец корпуса	Новый	Не утвержден			
OMZ-CC.T02.	ОЗП верхняя	Новый	Не утвержден			
OMZ-CC.T03.	ОЗП нижняя	Новый	Не утвержден			
OMZ-CC.T04.	Обечайка опорная	Новый	Не утвержден			
OMZ-CC.T05.	ОАЗ	Новый	Не утвержден			
OMZ-CC.T06.	Патрубок CAO3	Новый	Не утвержден			
OMZ-CC.T07.	Патрубок КИП	Новый	Не утвержден			
OMZ-CC.T08.	Кольцо ДУ850	Новый	Не			

Рисунок 9 — Формирование шаблона

Таблица с шаблонами включает в себя следующую информацию:

- код проекта;
- название проекта;
- степень проработки;
- статус;

- колонки возможных действий.

Кнопка «редактирование» открывает окно редактирования шаблона (рисунок 10).

Проект
Дробилка КСД-2200Гр-ДМ

Главный Модуль
Дробилка КСД-2200Гр-ДМ

Код задачи	Операция	Длительность, дн.	
A10	Начало работ	0	A20/ОН/0 X A30/ОН/0 X A080/ОН/0 X
A130/ОН/0 X A180/ОН/0 X A230/ОН/0 X			
A20	Окончание работ	0	
Модуль			
Комплект запасных частей (мех.часть)	Заказ аналог	Сборка	+ ↑ ↓ 🗑
	051140-014	99.1273.00-84	
A30	Обеспеченность	0.3	A40/ОН/0 X
A40	Детали в сварку	15	A50/ОН/-2 X
A50	Сварка	10	A60/ОН/-2 X
A60	Механообработка	30	A70/ОН/-2 X

ДОБАВИТЬ ОПЕРАЦИИ СОХРАНИТЬ ЗАКРЫТЬ

Рисунок 10 — Окно редактирования шаблона

Следующая кнопка открывает меню, из которого можно запустить процедуры нормализации, оптимизации, наполнения шаблона или запустить все эти процедуры поочередно.

Последняя кнопка «утверждение шаблона» запускает процедуру утверждения шаблона.

Формирование плана

Вкладка формирование плана также представляет собой таблицу, в которой представлена информация о производственных планах (рисунок 11).

Сверху находится кнопка «Создать новый проект», позволяющая пользователю создать новый производственный план. План создается абсолютно пустым.

При двойном щелчке по проекту открывается окно формирования плана (рисунок 12).

Формирование плана Создать новый проект			
Код проекта	Название проекта	Степень проработки	Статус
UZTM.2019.V11-2	Общий план	Новый	Не утвержден
UZTM.2019.V11-3	Общий план	Новый	Не утвержден
UZTM.2019.V12-1	Общий план	Новый	Не утвержден
UZTM.2019.V12-2	Общий план	Новый	Не утвержден
UZTM.2019.V12-3	Общий план	Новый	Не утвержден
UZTM.2019.V12-4	Общий план	Новый	Не утвержден
UZTM.2019.V13-1	Общий план	Новый	Не утвержден
UZTM.2019.V13-2	Общий план 2019 г. (без выравнивания)	Новый	Не утвержден
UZTM.2019.V13-3	Общий план 2019 г. (с выравниванием) верс. 09.11.18	Новый	Не утвержден
UZTM.2019.V13-4	Общий план 2019 г. (без выравнивания) верс. 04.12 с АГМК	Новый	Не утвержден
UZTM.2019.V13-5	Общий план 2019 г. (без выравнивания) верс. 04.12	Ошибка, работы не производятся	Не утвержден
UZTM.2019.V13-6	Общий план 2019 г. (без выравнивания) верс. 04.12 For Test	Выполнено, работы произведены без ошибок	Не утвержден

Рисунок 11 — Формирование плана

Формирование проекта "TestProject"

Модуль Дробилка	Заводской № машины № 1	Срок по договору 01.	Дробилка КСД-2200Гр-ДМ	↑ ↓ 🗑
Модуль Дробилка	Заводской № машины № 2	Срок по договору 20.	Дробилка КСД-2200Гр-ДМ	↑ ↓ 🗑
Модуль Экскаватор	Заводской № машины № 1	Срок по договору 21.	Экскаватор ЭКГ-12А	↓ 🗑

Код	Наименование
← UZTM.T33	Бандаж стальной (1 шт.)
← UZTM.T32	Валки 30 тонн и больше (1 шт.)
← UZTM.T30	Валки до 9 тонн (1 шт.)
← UZTM.T31	Валки от 10 до 30 т (1 шт.)
← UZTM.T21	Вибровозбудитель
← UZTM.T07	Дробилка ККД-1500/180
← UZTM.T05	Дробилка КМД-2200Т6-Д
← UZTM.T04	Дробилка КМД-3000Т2-ДП-М
← UZTM.T08	Дробилка КРД-900/100
← UZTM.T06	Дробилка КСД-2200Гр-ДМ

СОХРАНИТЬ

ЗАКРЫТЬ

Рисунок 12 — Окно формирования плана

Слева отображаются изделия, добавленные в план, после добавления изделия, необходимо ввести заводской номер машины, срок выполнения оп договору и выбрать тип изделия.

Справа находятся шаблоны, доступные для добавления в план выполнения.

Редактирование плана

Вкладка редактирование плана (рисунок 13) предназначена для редактирования шаблонов внутри производственного плана. Окно редактирования плана открывается по нажатию на кнопку «редактировать» (рисунок 14).

Редактирование плана				
UZTM.2019.V12-3	Общий план	Новый	Не утвержден	Редактировать
UZTM.2019.V12-4	Общий план	Новый	Не утвержден	Редактировать
UZTM.2019.V13-1	Общий план	Новый	Не утвержден	Редактировать
UZTM.2019.V13-2	Общий план 2019 г. (без выравнивания)	Новый	Не утвержден	Редактировать
UZTM.2019.V13-3	Общий план 2019 г. (с выравниванием) верс. 09.11.18	Новый	Не утвержден	Редактировать
UZTM.2019.V13-4	Общий план 2019 г. (без выравнивания) верс. 04.12 с АГМК	Новый	Не утвержден	Редактировать
UZTM.2019.V13-5	Общий план 2019 г. (без выравнивания) верс. 04.12	Ошибка, работы не производятся	Не утвержден	Редактировать
UZTM.2019.V13-6	Общий план 2019 г. (без выравнивания) верс. 04.12 For Test	Выполнено, работы произведены без ошибок	Не утвержден	Редактировать
UZTM.2019.V13-7	Общий план 2019 г. (без выравнивания) верс. 04.12 For Test Done	Выполнено, работы произведены без ошибок	Не утвержден	Редактировать
UZTM.2019.V13-7-TEST-1	Общий план for TEST	Выполнено, работы произведены без ошибок	Не утвержден	Редактировать
UZTM.2019.V666	Общий план 2019 г. (без выравнивания) верс. 04.12	Новый	Не утвержден	Редактировать

Рисунок 13 — Редактирование плана

Редактирование плана

A2540/ОН/0

Код задачи	Операция	Длительность, дн	Процент выполнения	
A2330	Сокращение работ	0	0	%

Модель: Комплект запасных частей (мех. часть) Заказ аналог Сборка

<input type="checkbox"/> Код задачи: A2340	Операция: Обеспеченность	Длительность, дн: 0,3	Процент выполнения: 0	A2350/ОН/0
<input type="checkbox"/> Код задачи: A2350	Операция: Детали в сборку	Длительность, дн: 15	Процент выполнения: 0	A2360/ОН/2
<input type="checkbox"/> Код задачи: A2360	Операция: Сварка	Длительность, дн: 10	Процент выполнения: 0	A2370/ОН/2
<input type="checkbox"/> Код задачи: A2370	Операция: Механообработка	Длительность, дн: 30	Процент выполнения: 0	A2380/ОН/2
<input type="checkbox"/> Код задачи: A2380	Операция: Сборка	Длительность, дн: 15	Процент выполнения: 0	A2330/ОН/0

Модель: (Поставляющиеся узлы) Заказ аналог (Плюс)

СОХРАНИТЬ ЗАКРЫТЬ

Рисунок 14 — Окно редактирования плана

Расчет плана

На вкладке расчет плана (рисунок 15) реализована возможность запуска расчетов планов производства с различными параметрами (рисунок 16).

Для расчета одновременно нескольких планов с одинаковыми параметрами, необходимо отметить нужные в левой части окна и нажать на кнопку «Расчет».

полагается форма запуска расчетов по отчету, а ниже таблица с доступными для скачивая отчетами.

Web-Primavera

https://workdurations.uralmash.ru/reports

Отчёты

Отчет по анализу трудоемкости

Название проекта: UZTM.2019.V13-1 Дата: 01.01.2019 Период: квартал **ОТЧЕТ**

Please select your currency

Отчет по материалам

Анализ технологических процессов по перегруженным ресурсам

Название проекта: UZTM.2019.V13-1 **ОТЧЕТ**

Доступные отчеты

Проект	Статус
UZTM.2019.V13-1	Выполнено

ОТЧЕТ

Рисунок 17 — Отчеты

Ресурсы

Вкладка ресурсы (рисунок 18) представляет собой таблицу со структурой дерева. Это наиболее удобный способ отображения, позволяющий увидеть, как именно группируются и объединяются ресурсы предприятия.

Ресурсы				
Тип	Идентификатор ресурса	Название ресурса	Не учитывать в ЦП	
▼ Корень ▼	UZTM-Оборудование	Станки, оборудование, машины	Нет	
▼ Объединение групп >	OMZCC	OMZCC	Нет	✎
▼ Группа оборудования	10006~5	1M63H-1.5	Да	
▼ Виды станков ▼	У-992	Протяжные станки	Нет	✎
▼ Объединение групп ▼	У-1008	ППГ-100	Нет	✎
▼ Группа оборудования	55086	ППГ-130	Да	
▼ Группа оборудования	55083	ППГ-100	Да	
▼ Объединение групп >	У-1009	ППГ-55	Нет	✎

Рисунок 18 — Ресурсы

Для каждого ресурса на предприятии введено понятие интенсивности, которое показывает доступность ресурса в определенное количество времени, например, часов в неделю (рисунок 19).

Группа оборудования	10006~5	1М63Н-1.5	Да	
Виды станков	У-992	Протяжные станки	Нет	
Действует с	Макс. интенсивность	Часов в день	Часов в неделю	+
01.01.2019	3.04583333	73.09999992	511.69999944	
10.10.2018	5	120	840	
05.10.2018	3.13888888	75.33333312	527.33333184	
01.01.2017	2.09444444	50.26666656	351.86666592	
Объединение групп	У-1008	ППГ-100	Нет	
Группа оборудования	55086	ППГ-130	Да	

Рисунок 19 — Интенсивность ресурса

Реализована возможность редактирования, добавления и удаления интенсивности ресурса.

3.5 Расчет экономической эффективности

3.5.1 Описание продукции

Создание web-интерфейса для системы циклового планирования для ПАО «Уралмашзавод» связано с материальными затратами на разработку и внедрение программного продукта. Анализ экономической эффективности является важнейшей задачей, он позволит оценить влияние внедряемого продукта на финансовое положение предприятия.

Эффективность внедрения программного продукта обуславливается действием ряда факторов организационного, информационного и экономического характера.

Организационный эффект проявляется в освобождении работников от операций по запуску наполнения, оптимизации и нормализации шаблонов и планов, а также расчета плана производства и формирование отчетов.

Экономический фактор проявляется в том, что вся обрабатываемая информация, в конечном счете, направлена на улучшение использования трудовых и временных ресурсов.

Базой для анализа экономической эффективности может служить время, затрачиваемое сотрудниками на формирование планов производства и запуск расчетов плана производства.

Оценить эффективность применения программного продукта можно с помощью прямых и косвенных показателей. Прямые (или экономические) показатели дают оценку автоматизации в денежном выражении, включают в себя определение затрат на разработку и эксплуатацию информационной системы (т.е. определение полной стоимости владения информационной системой), определение денежного потока, высвобожденного за счет информационной системы. К косвенным показателям можно отнести:

- повышение оперативности и актуальности информации;
- повышение качества информации, ее точности, детальности;
- снижение количества времени, затрачиваемого на обработку информации;
- повышение качества труда за счет сокращения рутинных операций.

3.5.2 Оценка затрат на разработку программного продукта

Учет фактических временных затрат представляем в виде таблицы 4.

Таблица 4 — Фактические временные затраты

Этапы разработки	Фактические затраты (час)
1. Подготовка и описание программного продукта	24
2. Исследование возможных средств реализации программного продукта	24
3. Разработка программного продукта	104
4. Тестирование и отладка программного продукта	40
5. Подготовка документации по продукту	24
ИТОГО	216

Коэффициент, учитывающий сложность разработки и затраты на корректировку исходного модуля определяем по формуле:

$$K_{сл} = c \cdot (1 + p), \quad (1)$$

где c — коэффициент сложности программы;

p — коэффициент коррекции программы в процессе разработки.

Таблица «Коэффициенты по категориям» (таблица 5) будет использована для расчета.

Таблица 5 — Коэффициенты по категориям

Наименование коэффициента	Категория			
	1	2	3	4
Коэффициент сложности программы	1,25	1,5	1,6	2,0
Коэффициент коррекции программы	0,05	0,1	0,5	1,0

$$K_{сл} = 1,25 (1+0,1) = 1,375$$

Затраты времени на разработку с учетом сложности программы выражаются произведением фактических временных затрат на разработку и коэффициента сложности.

$$t_{разр_{сл}} = t_{разр_{ф}} \cdot K_{сл}, \quad (2)$$

$$t_{разр_{сл}} = 216 \cdot 1,375 = 297 \text{ часов}$$

Расчет средней часовой оплаты программиста

Для определения средней часовой оплаты программиста определяем его годовой фонд заработной платы с учетом отчислений в социальные фонды (в Пенсионный фонд, Фонд обязательного медицинского страхования и Фонд социального страхования).

Месячный оклад стажера программиста в Уральском федеральном округе (УФО) 22 000 рублей, районный коэффициент для УФО составляет 15 %.

Учитывается процент премий (20 %). На суммарный фонд заработной платы начисляется процент взноса в социальные фонды. В 2019 году он со-

ставляет 30 %. Из него в Пенсионный фонд Российской Федерации (ПФР) — 22 %, Фонд социального страхования Российской Федерации (ФСС РФ) — 2,9 % и в Федеральный фонд обязательного медицинского страхования (ФФОМС) — 5,1 %.

Определяем месячный оклад программиста с учетом квалификации и всех надбавок.

$$ЗП_M = ЗП_{ОСН} \cdot (1+K_{ДОП}) \cdot (1+K_{УР}) \cdot (1+K_{СН}), \quad (3)$$

где $ЗП_{ОСН}$ — месячная оплата труда установленной квалификации;

$K_{ДОП}$ — коэффициент надбавок и премий;

$K_{УР}$ — уральский коэффициент;

$K_{СН}$ — коэффициент, учитывающий норму взноса в социальные фонды.

$$ЗП_M = 22\,000 \cdot (1+0,2) \cdot (1+0,15) \cdot (1+0,30) = 39\,468 \text{ руб.}$$

Годовой фонд заработной платы с учетом отчислений.

$$\Phi ЗР_Г = ЗП_M \cdot 12, \quad (4)$$

где 12 — количество месяцев в году.

$$\Phi ЗР_Г = 39\,468 \cdot 12 = 473\,616 \text{ руб.}$$

Число рабочих часов в году определяется согласно производственному календарю на 2019 г.

$$n_p = (N - N_{ПВ}) \cdot N_{СМ} - N_{ПП} \cdot 1, \quad (5)$$

где N — общее число дней в году;

$N_{ПВ}$ — число праздничных и выходных дней в году;

$N_{ПП}$ — число предпраздничных дней в году;

$N_{СМ}$ — продолжительность смены;

1 — величина сокращений предпраздничных рабочих дней.

Согласно производственного календаря на 2019 год продолжительность рабочего времени 1 970 часов при 40-часовой неделе.

Средняя часовая оплата программиста определяется соотношением

$$C_{разр} = \frac{\Phi ЗР_Г}{n_p}, \quad (6)$$

где $\Phi ЗР_{Г}$ — годовой фонд заработной платы с учетом отчислений;

n_p — число рабочих часов в году.

$$C_{разр} = \frac{473\,616}{1970} = 240,41 \text{ руб./час.}$$

Расходы по оплате труда разработчика программы определяются по следующей формуле:

$$З_{разр} = t_{разр.сл} \cdot C_{разр}, \quad (7)$$

где $t_{РАЗР.СЛ}$ — трудоемкость создания программы, с учетом сложности программы, выраженная в часах;

$C_{РАЗР}$ — средняя часовая оплата труда инженера-программиста.

$$З_{разр} = 297 \cdot 240,41 = 71\,401,77 \text{ руб.}$$

3.5.3 Расчет годового фонда времени работы на персональном компьютере

Определив действительный годовой фонд времени работы ПК в часах, получим возможность оценить себестоимость часа машинного времени. Время профилактики: ежедневно — 0.5 часа, ежемесячно — 2 часа, ежегодно — 16 часов.

$$n_{РПК} = n_p - N_{РЕМ}, \quad (8)$$

где N — общее число дней в году;

$N_{ПВ}$ — число праздничных и выходных дней в году;

$N_{ПП}$ — число предпраздничных дней в году;

$N_{СМ}$ — продолжительность смены;

1 — величина сокращений предпраздничных рабочих дней;

$N_{РЕМ}$ — время на проведение профилактических мероприятий.

$$n_{РПК} = 1970 - 153,5 = 1816,5 \text{ часов.}$$

$$N_{РЕМ} = (N - N_{П} - N_{В}) \cdot K_{Д} + K_{М} \cdot 12 + K_{Г}, \quad (9)$$

где $K_{Д}$ — коэффициент ежедневных профилактик (0.5);

$K_{М}$ — коэффициент ежемесячных профилактик (2);

12 — количество месяцев в году;

K_G — коэффициент ежегодных профилактик (6).

$$N_{PEM} = (365 - 118) \cdot 0,5 + 2 \cdot 12 + 6 = 153,5 \text{ часов.}$$

Годовые отчисления на амортизацию персонального компьютера

Балансовая стоимость персонального компьютера (ПК)

$$Ц_{ПК} = Ц_P \cdot (1 + K_{УН}), \quad (10)$$

где $Ц_P$ — рыночная стоимость ПК (определяется по прайсу);

$K_{УН}$ — коэффициент, учитывающий затраты на установку и наладку.

$$Ц_{ПК} = 45\,000 (1 + 0,1) = 49\,500 \text{ руб.}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$З_{ГАМ} = Ц_{ПК} \cdot H_A, \quad (11)$$

где $Ц_{ПК}$ — балансовая стоимость ПК;

H_A — норма амортизационных отчислений за год.

$$З_{ГАМ} = 49\,500 \cdot 0,2 = 9\,900 \text{ руб.}$$

$$H_a = \frac{1}{T \frac{ПК}{ЭКС}} \cdot 100, \quad (12)$$

где $T \frac{ПК}{ЭКС}$ — полезный срок действия ПК.

$$H_a = \frac{1}{5} \cdot 100 = 20\% .$$

Затраты на электроэнергию

$$З_{ГЭЛ} = P_{ЧПК} \cdot T_{ГПК} \cdot Ц_{ЭЛ} \cdot K_{ИНТ}, \quad (13)$$

где $P_{ЧПК}$ — установочная мощность ПК;

$T_{ГПК}$ — годовой фонд полезного времени работы машины ($n_{ПК}$);

$Ц_{ЭЛ}$ — стоимость 1 кВт/ч. электроэнергии ($Ц_{ЭЛ}=3,7$ руб./кВт/ч);

$K_{ИНТ}$ — коэффициент интенсивного использования ПК (0,9).

$$З_{ГЭЛ} = 0,35 \cdot 1816,5 \cdot 3,7 \cdot 0,9 = 2\,117,13 \text{ руб.}$$

Текущие затраты на эксплуатацию ПК рассчитываются по формуле:

$$З_{ГПК} = З_{ГАМ} + З_{ГЭЛ}, \quad (14)$$

где $З_{ГАМ}$ — годовые отчисления на амортизацию;

$З_{ГЭЛ}$ — годовые затраты на электроэнергию для компьютера.

$$З_{ГПК} = 9\,900 + 2\,117,13 = 12\,017,13 \text{ руб.}$$

Себестоимость часа работы на компьютере:

$$С_{ПК} = \frac{З_{ГПК}}{n_{РПК}}, \quad (15)$$

где $З_{ГПК}$ — годовые затраты на ПК;

n

$_{РПК}$ — годовой фонд полезного времени работы машины.

$$С_{ПК} = \frac{12017,13}{1816,5} = 6,62 \text{ руб./час}$$

Трудоемкость использования компьютера

В ходе разработки программного продукта, машина используется на этапах программирования по готовой блок-схеме алгоритма, отладки программы на компьютере, подготовки документации по задаче.

Совокупные затраты машинного времени составляют:

$$t_{МАШ} = (t_{ПФ} + t_{ОТЛФ} + t_{ДФ}) \cdot K_{СЛ}, \quad (16)$$

$$t_{МАШ} = (104 + 40 + 24) \cdot 1,375 = 231 \text{ час.}$$

Затраты на оплату машинного времени

Затраты на оплату машинного времени рассчитываются по формуле:

$$З_{МАШ} = t_{МАШ} \cdot С_{ПК}, \quad (17)$$

$$З_{МАШ} = 231 \cdot 6,62 = 1\,529,22 \text{ руб.}$$

Общие затраты на создание программы

Общие затраты на создание программы определяем как сумма затрат на разработку программы и затрат на оплату машинного времени.

$$З_{ОБЩ} = З_{РАЗР} + З_{МАШ}, \quad (18)$$

$$З_{ОБЩ} = 71\,401,77 + 1\,529,22 = 72\,930,99 \text{ руб.}$$

Общие затраты на расходные материалы

При формировании общих затрат на создание программного продукта, кроме затрат на оплату труда разработчика и на оплату машинного времени, необходимо учитывать затраты, связанные с использованием расходных ма-

териалов и комплектующих в процессе проектирования, разработки и внедрения. Статьи затрат на расходные материалы с указанием их стоимости показаны в таблице 6.

Таблица 6 — Расходные материалы

Статьи затрат	Стоимость за единицу	Количество	Общая стоимость
1. Пользование ресурсами Интернет	1,18 руб./час	50 ч.	59 руб.
2. Бумага	0,51 руб./л.	20 л.	10,20 руб.
Итого: (З _{РМ})			69,20 руб.

Общие затраты на разработку программного продукта приведены в таблице 7.

Таблица 7 — Общие затраты на разработку программного продукта

Статьи затрат	Условное обозначение	Числовое значение
1. Общие затраты на заработную плату	З _{ОБЩ}	72930,99 руб.
2. Расходные материалы	З _{РМ}	69,20 руб.
Итого: (С _{ОБЩ})		73000,19 руб.

3.5.4 Предполагаемая цена программного продукта с учетом нормы прибыли

Вычислим предполагаемую цену продукта с учетом нормы прибыли:

$$Ц = С_{ОБЩ} \cdot (1+N), \quad (19)$$

где $С_{ОБЩ}$ — общие затраты на разработку программного продукта;

N — норма ожидаемой прибыли.

$$Ц = 73000,19 \cdot (1+0,2) = 87600,23 \text{ руб.}$$

Предполагаемая цена данного продукта с учетом нормы прибыли составила 88000 руб.

3.5.5 Расчет экономической эффективности программы

Результатом внедрения в учреждении данного программного продукта будет сокращение трудовых и временных ресурсов на обслуживание заявок сотрудников.

В таблице 8 указаны показатели, помогающие рассчитать эффективность от внедрения данного программного продукта.

Таблица 8 — Расчет экономической эффективности

Показатель сотрудников	Единицы измерения	До внедрения системы	После внедрения системы
Время на запуск процедур расчета плана, нормализации, оптимизации и наполнения	час.	0,5	0,017
Среднее количество запусков месяц	шт.	100	100
Временные затраты в месяц	час.	50	1,7
Сэкономленное время	час.	-	48,3
Средняя заработная плата сотрудников в час	руб.	300	300
Стоимость сэкономленных временных затрат в месяц	руб.	-	14 490
Время на формирование различных отчётов	мин.	90	3
Среднее количество отчетов в месяц	шт.	35	35
Временные затраты в месяц	час.	52,5	1,75
Сэкономленное время	час.	-	50,75
Средняя заработная плата сотрудников в час	руб.	250	250
Стоимость сэкономленных временных затрат в месяц	руб.	-	12 687,50
Общая стоимость сэкономленных временных затрат в месяц	руб.		27 177,50

Следовательно, учитывая, что предполагаемая стоимость программного продукта составляет 88000 рублей, найдем срок его окупаемости.

$$T_{\text{окуп}} = \frac{88000}{27\,177,50} = 3,25 \text{ мес.}$$

Полученный программный продукт обладает достаточно быстрой окупаемостью.

ЗАКЛЮЧЕНИЕ

Web-интерфейс для системы циклового планирования будет служить на ПАО «Уралмашзавод» инструментом запуска расчета циклового плана, процедур наполнения шаблонов, нормализации, оптимизации и утверждения шаблонов изделий и планов производства, инструментом формирования планов производства, что повысит эффективность работы отдела планирования и снизит нагрузку на администратора базы данных

В соответствии с поставленными задачами:

1. Проведен анализ литературы и интернет-источников, посвященных проблемам разработки и внедрения информационных систем, процессам в области планирования производства.

2. Изучены и проанализированы основные процессы предметной области. К основной задаче отдела планирования ПАО «Уралмашзавод» можно отнести оптимизацию загрузки оборудования предприятия, с целью выполнения производственных сроков по изготовлению продукции.

3. Изучены возможности и технологические решения Primavera Р6. Определены требования к программному продукту, разработан проект системы. Система должна обладать высокой скоростью обработки данных, гибкостью, надёжностью.

4. Обоснована необходимость разработки web-интерфейса. Разработан программный продукт позволяющий автоматизировать процессы запуска расчетов циклового плана, нормализации и оптимизации шаблонов изделий и производственных планов и пр.

5. Рассчитана экономическая эффективность внедрения разработанной конфигурации.

Таким образом, все задачи были выполнены, цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Автоматизация параметрического моделирования в машиностроительном производстве [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/avtomatizatsiya-parametricheskogo-modelirovaniya-v-mashinostroitelnom-proizvodstve> (дата обращения: 20.05.2019).
2. Анализ технологии и разработка методов повышения энергоэффективности производства [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/analiz-tehnologii-i-razrabotka-metodov-povysheniya-energoeffektivnosti-proizvodstva> (дата обращения: 20.05.2019).
3. Вестник удмуртского университета [Электронный ресурс]. — Режим доступа: http://en.vestnik.udsu.ru/files/originsl_articles/vuu_13_022_09.pdf (дата обращения: 01.05.2019).
4. ГОСТ 19.101-77. Виды программ и программных документов (с Изменением N 1) [Электронный ресурс]. — Введ. 01.01.1980. — Режим доступа: <http://docs.cntd.ru/document/1200007627> (дата обращения: 17.05.2019).
5. ГОСТ 19.102-77 ЕСПД. Стадии разработки [Электронный ресурс]. — Введ. 01.01.1980. — Режим доступа: <http://docs.cntd.ru/document/gost-19-102-77> (дата обращения: 17.05.2019).
6. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам [Электронный ресурс]. — Введ. 01.01.1980. — Режим доступа: <http://docs.cntd.ru/document/gost-19-105-78-espd> (дата обращения: 17.05.2019).
7. Интегрированная система автоматизации проектирования технологических процессов и оперативного управления производством [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/integrirovannaya-sistema-avtomatizatsii-proektirovaniya-tehnologicheskikh-protsessov-i-operativnogo-upravleniya-proizvodstvom> (дата обращения: 20.05.2019).
8. Исследование и сравнение современных реализаций flux-архитектур разработки web-приложений [Электронный ресурс]. — Режим

доступа: <https://cyberleninka.ru/article/n/issledovanie-i-sravnenie-sovremennyh-reali-zatsiy-flux-arhitektur-razrabotki-veb-prilozheniy> (дата обращения: 22.05.2019).

9. Исследование методов увеличения производительности web-приложений [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/issledovanie-metodov-uvelicheniya-proizvoditelnosti-web-prilozheniy> (дата обращения: 05.05.2019).

10. Модель экономической эффективности автоматизации конструкторско-технологической подготовки машино-строительного производства [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/model-ekonomicheskoy-effektivnosti-avtomatizatsii-konstruktorsko-tehnologicheskoy-podgotovki-mashino-stroitel'nogo-proizvodstva> (дата обращения: 20.05.2019).

11. Оперативно-календарное планирование в серийном и единичном производстве [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/operativno-kalendarnoe-planirovanie-v-seriynom-i-edinichnom-proizvodstve> (дата обращения: 03.05.2019).

12. Особенности и методология производственного планирования на машиностроительном предприятии [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/osobennosti-i-metodologiya-proizvodstvennogo-planirovaniya-na-mashinostroitel'nom-predpriyatii> (дата обращения: 30.04.2019).

13. Официальный сайт Golang [Электронный ресурс]. — Режим доступа: <https://golang.org/> (дата обращения: 22.05.2019).

14. Официальный сайт Oracle [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/ru/applications/primavera/solutions/products.html> (дата обращения: 30.04.2019).

15. Официальный сайт React.js [Электронный ресурс]. — Режим доступа: <https://reactjs.org/> (дата обращения: 22.05.2019).

16. Официальный сайт ПАО «Уралмашзавод» [Электронный ресурс]. — Режим доступа: <https://www.uralmash.ru/> (дата обращения: 12.05.2019).

17. Планирование производства продукции в условиях конкурентной среды [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/planirovanie-proizvodstva-produktsii-v-usloviyah-konkurentnoy-sredy> (дата обращения: 03.05.2019).

18. Почему стоит использовать React JS при разработке приложений [Электронный ресурс]. — Режим доступа: <https://xbsoftware.ru/blog/pochemu-stoit-ispolzovat-react-js-razrabotke-prilozhenij/> (дата обращения: 22.05.2019).

19. Пресс-центр ПАО «Уралмашзавод» [Электронный ресурс]. — Режим доступа: <https://www.uralmash.ru/press-center/news/4991/> (дата обращения: 07.05.2019).

20. Принципы стратегического планирования на предприятии [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/printsipy-strategicheskogo-planirovaniya-na-predpriyatii-1> (дата обращения: 07.05.2019).

21. Программирование, стартапы, новости IT [Электронный ресурс]. — Режим доступа: <https://techrocks.ru/2018/07/29/programming-languages-popularity-2018/> (дата обращения: 22.05.2019).

22. Разработка web-приложений для поддержки стратегического управления [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/razrabotka-web-prilozheniy-dlya-podderzhki-strategicheskogo-upravleniya> (дата обращения: 05.05.2019).

23. Разработка web-сервиса для автоматизированной генерации документов на основе docx-шаблонов [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/razrabotka-web-servisa-dlya-avtomatizirovannoy-generatsii-dokumentov-na-osnove-docx-shablonov> (дата обращения: 22.05.2019).

24. Разработка программного обеспечения [Электронный ресурс]. — Режим доступа: <https://xbsoftware.ru/blog/pochemu-stoit-ispolzovat-react-js-razrabotke-prilozhenij/> (дата обращения: 12.05.2019).

25. Распределение последовательности подзадач при календарном планировании производства [Электронный ресурс]. — Режим доступа:

<https://cyberleninka.ru/article/n/raspredelenie-posledovatelnosti-podzadach-pri-kalendarnom-planirovanii-proizvodstva> (дата обращения: 03.05.2019).

26. Сайт о программировании METANIT.COM [Электронный ресурс]. — Режим доступа: <https://metanit.com/sql/sqlserver/1.1.php> (дата обращения: 22.05.2019).

27. Современные интерактивные web-приложения — построение пользовательского интерфейса с react [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/sovremennye-interaktivnye-veb-prilozheniya-postroenie-polzovatelskogo-interfeysa-s-react> (дата обращения: 22.05.2019).

28. Технологии создания пользовательских web-приложений [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-sozdaniya-polzovatelskih-web-prilozheniy> (дата обращения: 05.05.2019).

29. Энциклопедия экономиста Grandars [Электронный ресурс]. — Режим доступа: <http://www.grandars.ru/student/ekonomicheskaya-teoriya/tip-proizvodstva.html> (дата обращения: 07.05.2019).

30. Язык программирования Go: мнения и перспектива [Электронный ресурс]. — Режим доступа: <https://timeweb.com/ru/community/articles/yazyk-programirovaniya-go-mneniya-i-perspektiva-1> (дата обращения: 13.05.2019).

31. Economic-S [Электронный ресурс]. — Режим доступа: <http://www.economic-s.ru/index.php/theory/operativnoe-planirovanie-opp/metodyi-kalendarnogo-planirovaniya-pr/> (дата обращения: 07.05.2019).

32. Hardline.ru [Электронный ресурс]. — Режим доступа: <http://hardline.ru/2/13/995/> (дата обращения: 02.05.2019).

33. StackOverflow [Электронный ресурс]. — Режим доступа: <https://ru.stackoverflow.com/> (дата обращения: 12.05.2019).

34. Web-приложения и данные: проблемы абстракции и масштабируемости [Электронный ресурс]. — Режим доступа: <https://cyberleninka.ru/article/n/web-prilozheniya-i-dannye-problemy-abstraktsii-i-masshtabiruемости> (дата обращения: 30.04.2019).

ПРИЛОЖЕНИЕ А

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»**

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
Направление подготовки 09.03.03 Прикладная информатика
Профиль «Прикладная информатика в экономике»

УТВЕРЖДАЮ
Заведующий кафедрой

И. А. Суслова

подпись

и.о. фамилия

« ____ » _____ 201_ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студента (ки) _____ 4 _____ курса группы _____ ИЭ-402п
Куимовой Анастасии Сергеевны
фамилия, имя, отчество полностью

1. Тема **Web-приложение для системы циклового планирования на основе Oracle Primavera Р6**

утверждена распоряжением по институту от « ____ » _____ 20 ____ г. № ____

2. Руководитель _____ Ченушкина Светлана Владимировна
фамилия, имя, отчество полностью

_____ ст. преподаватель _____ РГППУ
ученая степень _____ ученое звание _____ должность _____ место работы _____

3. Место преддипломной практики
ФГАОУ ВО «Российский государственный профессионально-педагогический университет», г. Екатеринбург

4. Исходные данные к ВКР _____
Официальный сайт Golang [Электронный ресурс]. — Режим доступа: <https://golang.org/>, Официальный сайт React.js [Электронный ресурс]. — Режим доступа: <https://reactjs.org/>

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)

Анализ различных источников по теме

Анализ предметной области

Обоснование проектных решений

Описание реализации

Расчет экономической эффективности

6. Перечень демонстрационных материалов _____ Презентация в MS Power Point

7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа дипломной работы	Срок выполне- ния этапа	Процент выполне- ния ВКР	Отметка руково- дителя о выполне- нии
1	Сбор информации по выпускной квалификационной работе	03.05.2019	10 %	
2	Выполнение работ по разрабатываемым вопросам и их изложение в пояснительной записке:		65 %	
2.1	Анализ различных источников по теме	08.05.2019	15 %	
2.2	Анализ предметной области	14.05.2019	15 %	
2.3	Проектирование	20.05.2019	15 %	
2.4	Реализация прикладного решения	29.05.2019	15 %	
2.5	Расчет экономической эффективности	31.05.2019	5 %	
3	Оформление текстовой части ВКР	04.06.2019	10 %	
4	Нормоконтроль	07.06.2019	10 %	
5	Подготовка доклада к защите в ГЭК	10.06.2019	5 %	

8. Консультанты по разделам выпускной квалификационной работы

Наименование раздела	Консультант	Задание выдал		Задание принял	
		подпись	дата	подпись	дата

Руководитель _____
подпись дата

Задание получил _____
подпись студента дата

9. Выпускная квалификационная работа и все материалы проанализированы.

Считаю возможным допустить Куимову А. С. к защите выпускной квалификационной работы в государственной экзаменационной комиссии.

Руководитель _____
подпись дата

10. Допустить Куимову А. С. к защите выпускной квалификационной работы
фамилия и. о. студента

в государственной экзаменационной комиссии (протокол заседания кафедры
от «__» _____ 20__ г., № _____)

Заведующий кафедрой _____
подпись дата

ПРИЛОЖЕНИЕ Б

Основа интерфейса, клиентская часть:

```
import React from 'react'
import { Route, Link, Redirect, withRouter } from 'react-router-dom'
import { withStyles } from '@material-ui/core/styles'
import CssBaseline from '@material-ui/core/CssBaseline'
import AppBar from '@material-ui/core/AppBar'
import Toolbar from '@material-ui/core/Toolbar'
import Button from '@material-ui/core/Button'
import Paper from '@material-ui/core/Paper'
import TextField from '@material-ui/core/TextField'
import Grid from '@material-ui/core/Grid'
import 'typeface-roboto'
import $ from 'jquery'
import PropTypes from 'prop-types'
import MakeTemplate from './MakeTemplate/MakeTemplate'
import AssertedTemplate from './AssertedTemplate/AssertedTemplate'
import MakePlan from './MakePlan/MakePlan'
import EditPlan from './EditPlan/EditPlan'
import CalcPlan from './CalcPlan/CalcPlan'
import Reports from './Reports/Reports'
import Rsrc from './Rsrc/Rsrc'
import Desktop from './Desktop/Desktop'
import Drawer from '@material-ui/core/Drawer';
import List from '@material-ui/core/List';
import Typography from '@material-ui/core/Typography';
import Divider from '@material-ui/core/Divider';
import IconButton from '@material-ui/core/IconButton';
import MenuIcon from '@material-ui/icons/Menu';
import ChevronLeftIcon from '@material-ui/icons/ChevronLeft';
import ChevronRightIcon from '@material-ui/icons/ChevronRight';
import ListItem from '@material-ui/core/ListItem';
import ListItemText from '@material-ui/core/ListItemText';
import classNames from 'classnames';
import CreateTemplate from './MakeTemplate/CreateTemplate'
import CreateProject from './MakePlan/CreateProject'
import GranttDialog from './Desktop/GranttDialog'
import AcceptDialog from './Desktop/AcceptDialog'
import MuiThemeProvider from '@material-ui/core/styles/MuiThemeProvider';
import theme from './theme';

const drawerWidth = 240;

const styles = theme => ({
  .....
})

const myAuth = {
  isAuthenticated: false,
  isAdmin: false,
  username: '',
  authenticate(user, pass, cb) {
    $.ajax({
      url: '/api/auth',
      dataType: 'json',
      contentType: 'application/json;charset=utf-8',
      method: 'POST',
```

```

    data: JSON.stringify({ username: user, password: pass }),
    success: function (username) {
      this.isAuthenticated = true
      this.username = username
      if (cb) cb(username)
    }.bind(this)
  })
},
logout(cb) {
  this.isAuthenticated = false
  this.username = ""
  $.ajax({
    url: '/api/logout',
    dataType: 'json',
    contentType: 'application/json;charset=utf-8',
    method: 'GET',
    success: function () {
      window.location = "/login"
      if (cb) cb()
    }
  })
},
check(cb) {
  $.ajax({
    url: '/api/autz',
    dataType: 'json',
    contentType: 'application/json;charset=utf-8',
    method: 'GET',
    success: function (data) {
      this.isAuthenticated = true;
      this.username = data.username
      this.isAdmin = data.isAdmin
      if (cb) cb(data)
    }.bind(this)
  })
}
};

```

```

function PrivateRoute({ component: Component, ...rest }) {
  return (
    <Route
      {...rest}
      render={props =>
        myAuth.isAuthenticated ? (
          <Component {...props} />
        ) : (
          <Redirect
            to={{
              pathname: "/login",
              state: { from: props.location }
            }}
          />
        )
      }
    />
  )
}

```

```

class App extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      isLoggedIn: false,

```



```

    isAdmin: false,
    username: "",
    open: false,
  }
}

handleDrawerOpen = () => {
  this.setState({ open: true });
};

handleDrawerClose = () => {
  this.setState({ open: false });
};

handleLogout = () => {
  myAuth.signout() => {
  })
}

componentDidMount() {
  myAuth.check()
}

getHeadline(param) {
  switch (param) {
    case '/maketemplate':
      return 'Формирование шаблона';
    case '/asserted':
      return 'Утвержденные шаблоны';
    case '/makeplan':
      return 'Формирование плана';
    case '/editplan':
      return 'Редактирование плана';
    case '/calcplan':
      return 'Расчёт плана';
    case '/reports':
      return 'Отчёты';
    case '/rsrc':
      return 'Ресурсы';
    case '/desktop':
      return 'Рабочий стол (beta version)';
    default:
      return 'Web-Primavera';
  }
}

render() {
  const { classes } = this.props;
  const { open } = this.state;
  return (
    <MuiThemeProvider theme={theme}>
      <React.Fragment>
        <CssBaseline />
        <AppBar
          position="fixed"
          className={classNames(classes.appBar, {
            [classes.appBarShift]: open,
          })}
        >
          <Toolbar disableGutters={!open}>
            {(myAuth.isAuthenticated) &&
              <IconButton
                color="inherit"

```

```

        aria-label="Open drawer"
        onClick={this.handleDrawerOpen}
        className={classNames(classes.menuButton, open && classes.hide)}
      >
        <MenuIcon />
      </IconButton>
    }
    <Typography variant="h6" color="inherit" className={classes.button}>
      {this.getHeadline(window.location.pathname)}
    </Typography>

    {window.location.pathname === "/maketemplate" && <CreateTemplate />}
    {window.location.pathname === "/makeplan" && <CreateProject />}

    {window.location.pathname === "/desktop" && <GranttDialog Item_id={666} Item_name=""
Item_level={1} />}
    {window.location.pathname === "/desktop" && <AcceptDialog />}

  </Toolbar>
</AppBar>
<Drawer
  className={classes.drawer}
  anchor="left"
  open={open}
  onClose={this.handleDrawerClose}

  classes={{
    paper: classes.drawerPaper,
  }}
>
  <div className={classes.drawerHeader}>
    <IconButton onClick={this.handleDrawerClose}>
      {open ? <ChevronLeftIcon /> : <ChevronRightIcon />}
    </IconButton>
  </div>
  <Divider />
  <List>

    <ListItem button value="/maketemplate" component={Link} to="/maketemplate" on-
Click={this.handleDrawerClose}>
      <ListItemText>Формирование шаблона</ListItemText>
    </ListItem>
    <ListItem button value="/asserted" component={Link} to="/asserted" on-
Click={this.handleDrawerClose}>
      <ListItemText>Утверждённые шаблоны</ListItemText>
    </ListItem>
    <ListItem button value="/makeplan" component={Link} to="/makeplan" on-
Click={this.handleDrawerClose}>
      <ListItemText>Формирование плана</ListItemText>
    </ListItem>
    <ListItem button value="/editplan" component={Link} to="/editplan" on-
Click={this.handleDrawerClose}>
      <ListItemText>Редактирование плана</ListItemText>
    </ListItem>
    <ListItem button value="/calcplan" component={Link} to="/calcplan" on-
Click={this.handleDrawerClose}>
      <ListItemText>Расчёт плана</ListItemText>
    </ListItem>
    <ListItem button value="/reports" component={Link} to="/reports" on-
Click={this.handleDrawerClose}>
      <ListItemText>Отчёты</ListItemText>
    </ListItem>
  </List>

```

```

        <ListItem button value="/rsrc" component={Link} to="/rsrc" on-
Click={this.handleDrawerClose}>
        <ListItemText>Ресурсы</ListItemText>
        </ListItem>

        <ListItem button value="/desktop" component={Link} to="/desktop" on-
Click={this.handleDrawerClose}>
        <ListItemText>Рабочий стол</ListItemText>
        </ListItem>

        <Divider />
        <ListItem button onClick={this.handleLogout} component={Link} to="/login" >
        <ListItemText>Выход</ListItemText>
        </ListItem>

    </List>

</Drawer>
<Route exact path="/" component={() => <Redirect to="/maketemplate" />} />
<PrivateRoute path="/maketemplate" component={MakeTemplate} />
<PrivateRoute path="/asserted" component={AssertedTemplate} />
<PrivateRoute path="/makeplan" component={MakePlan} />
<PrivateRoute path="/editplan" component={EditPlan} />
<PrivateRoute path="/calcplan" component={CalcPlan} />
<PrivateRoute path="/reports" component={Reports} />
<PrivateRoute path="/rsrc" component={Rsrc} />
<PrivateRoute path="/desktop" component={Desktop} />

    <Route path="/login" component={AuthForm} />
</React.Fragment>
</MuiThemeProvider>
)
}
}

App.propTypes = {
  classes: PropTypes.any,
  history: PropTypes.any,
}

export default withRouter(withStyles(styles)(App))

const authstyles = theme => ({
  authform: {
    width: '400',
    marginTop: theme.spacing.unit * 8
  },
  textField: {
    marginLeft: theme.spacing.unit,
    marginRight: theme.spacing.unit,
    width: '95%'
  },
  paper: {
    marginTop: 16,
    padding: 16
  },
  button: {
    margin: theme.spacing.unit,
  }
})

```

```

class AuthFormBase extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      username: "",
      password: "",
      redirectToReferrer: false
    }
  }

  componentDidMount() {
    myAuth.check() => {
      this.setState({ redirectToReferrer: true })
    })
  }

  handleChange = event => {
    this.setState({ [event.target.name]: event.target.value });
  }

  handleSubmit = event => {
    myAuth.authenticate(this.state.username, this.state.password, () => {
      this.setState({ redirectToReferrer: true })
    })
    event.preventDefault()
  }

  render() {
    const { classes } = this.props
    let { from } = this.props.location.state || { from: { pathname: "/" } }
    if (this.state.redirectToReferrer) return <Redirect to={from} />
    return (
      <MuiThemeProvider theme={theme}>
        <Grid container justify="center" alignItems="center" direction="column" class-
Name={classes.authform} spacing={16}>
          <form noValidate autoComplete="off">
            <Paper className={classes.paper}>
              <h1>Требуется авторизация</h1>
              <Grid item xs={12}>
                <TextField name="username" id="username" label="Имя пользователя" margin="normal"
className={classes.textField}
                value={this.state.username}
                onChange={this.handleChange}
                />
              </Grid>
              <Grid item xs={12}>
                <TextField name="password" id="password" label="Пароль" type="password" autoCom-
plete="current-password" margin="normal" className={classes.textField}
                value={this.state.password}
                onChange={this.handleChange}
                />
              </Grid>
              <Grid item xs={12}>
                <Button type="submit" variant="contained" color="primary" className={classes.button} on-
Click={this.handleSubmit} autoFocus>
                  Войти
                </Button>
              </Grid>
            </Paper>
          </form>
        </Grid>
      </MuiThemeProvider>
    );
  }
}

```

```

}
}

```

```

AuthFormBase.propTypes = {
  classes: PropTypes.any,
  handleAuth: PropTypes.func,
  location: PropTypes.object,

```

```

}

```

```

const AuthForm = withStyles(authstyles)(AuthFormBase)

```

Основа интерфейса, серверная часть:

```

package main

```

```

import (
    "database/sql"
    "fmt"
    "log"
    "net/http"

    "git.uralmash.ru/planningOfWorkDurations/app/report1"
    _ "github.com/denisenkom/go-mssqldb"
    "github.com/rs/cors"
)

```

```

const connectionString = "....."

```

```

type jwtContextKey string

```

```

func errorWithJSON(w http.ResponseWriter, message string, code int) {
    w.Header().Set("Content-Type", "application/json; charset=utf-8")
    // w.Header().Set("Access-Control-Allow-Origin", "*")
    w.WriteHeader(code)
    fmt.Fprintf(w, `{"message": %q}`, message)
}

```

```

func responseWithJSON(w http.ResponseWriter, json []byte, code int) {
    w.Header().Set("Content-Type", "application/json; charset=utf-8")
    // w.Header().Set("Access-Control-Allow-Origin", "*")
    w.WriteHeader(code)
    if _, err := w.Write(json); err != nil {
        log.Println(err)
    }
}

```

```

func main() {
    log.SetFlags(log.LstdFlags | log.Lshortfile)

    db, err := sql.Open("sqlserver", connectionString)
    if err != nil {
        log.Fatal(err)
    }
    defer db.Close()

    c := cors.New(cors.Options{
        AllowedOrigins: []string{"*"},
        AllowedHeaders: []string{"Authorization", "Content-Type"},
        AllowCredentials: true,
    })

    mux := http.NewServeMux()

```

```

// makeTemplate.go
mux.HandleFunc("/api/getProjects", validateMiddleware(getProjects(db)))
mux.HandleFunc("/api/getProjectDetails", validateMiddleware(getProjectDetails(db)))
mux.HandleFunc("/api/storeProjectDetails", validateMiddleware(storeProjectDetails(db)))
mux.HandleFunc("/api/getOrders", validateMiddleware(getOrders(db)))
mux.HandleFunc("/api/getArticles", validateMiddleware(getArticles(db)))
mux.HandleFunc("/api/fillWorks", validateMiddleware(fillWorks(db)))
mux.HandleFunc("/api/reloadRsrc", validateMiddleware(reloadRsrc(db)))
mux.HandleFunc("/api/acceptTemplate", validateMiddleware(acceptTemplate(db)))
mux.HandleFunc("/api/createTemplate", validateMiddleware(createTemplate(db)))
mux.HandleFunc("/api/NormalizationTemplate", validateMiddle-
ware(NormalizationTemplate(db)))
mux.HandleFunc("/api/OptimizationTemplate", validateMiddleware(OptimizationTemplate(db)))
mux.HandleFunc("/api/ComplexTemplate", validateMiddleware(ComplexTemplate(db)))

// launchPlanning.go
mux.HandleFunc("/api/getLaunchProjects", validateMiddleware(getLaunchProjects(db)))
mux.HandleFunc("/api/runLaunchProjects", validateMiddleware(runLaunchProjects(db)))
mux.HandleFunc("/api/productionPlanCreate", validateMiddleware(productionPlanCreate(db)))
mux.HandleFunc("/api/getParams", validateMiddleware(getParams(db)))
mux.HandleFunc("/api/goProjectsWithParams", validateMiddleware(goProjectsWithParams(db)))

// makeProject.go
mux.HandleFunc("/api/getMakeProjects", validateMiddleware(getMakeProjects(db)))
mux.HandleFunc("/api/getMakeProjectDetails", validateMiddleware(getMakeProjectDetails(db)))
mux.HandleFunc("/api/getMakeProjectTemplates", validateMiddle-
ware(getMakeProjectTemplates(db)))
mux.HandleFunc("/api/storeMakeProjects", validateMiddleware(storeMakeProjects(db)))
mux.HandleFunc("/api/createProject", validateMiddleware(createProject(db)))
mux.HandleFunc("/api/getProductType", validateMiddleware(getProductType(db)))

// editPlan.go
mux.HandleFunc("/api/getPlans", validateMiddleware(getPlans(db)))
mux.HandleFunc("/api/projectPRSRCGroup", validateMiddleware(projectPRSRCGroup(db)))

// report1.go
mux.HandleFunc("/api/getReport1", validateMiddleware(report1.GetReport(db)))
mux.HandleFunc("/api/getReportMat", validateMiddleware(report1.GetReportMat(db)))
mux.HandleFunc("/api/runReportMat", validateMiddleware(report1.RunReportMat(db)))
mux.HandleFunc("/api/getReportOGT", validateMiddleware(report1.GetReportOGT(db)))
mux.HandleFunc("/api/setReportOGT", validateMiddleware(report1.SetReportOGT(db)))
mux.HandleFunc("/api/runReportOGT", validateMiddleware(report1.RunReportOGT(db)))
mux.HandleFunc("/api/getReports", validateMiddleware(report1.GetReports(db)))
//mux.HandleFunc("/api/getReport1", report1.GetReport(db))

// rsrc.go
mux.HandleFunc("/api/getRsrcList", validateMiddleware(getRsrcList(db)))
mux.HandleFunc("/api/getRsrcDetails", validateMiddleware(getRsrcDetails(db)))
mux.HandleFunc("/api/setRsrcList", validateMiddleware(setRsrcList(db)))
mux.HandleFunc("/api/setRsrcDetails", validateMiddleware(setRsrcDetails(db)))

// desktop.go
mux.HandleFunc("/api/getDesktopList", validateMiddleware(getDesktopList(db)))
mux.HandleFunc("/api/getDesktopSnack", validateMiddleware(getDesktopSnack(db)))
mux.HandleFunc("/api/getDesktopItemPercent", validateMiddle-
ware(getDesktopItemPercent(db)))
mux.HandleFunc("/api/getOrdersList", validateMiddleware(getOrdersList(db)))
mux.HandleFunc("/api/setOrder", validateMiddleware(setOrder(db)))

// auth.go
mux.HandleFunc("/api/auth", createTokenEndpoint(db))
mux.HandleFunc("/api/autz", checkAuthorization(db))

```

```

        mux.HandleFunc("/api/logout", logout)

        handler := c.Handler(mux)
        log.Fatal(http.ListenAndServe(":8888", handler))
    }

```

Пример отображения таблицы и загрузки данных, клиентская часть

```

class SelectTemplate extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      projects: [],
      addDialogOpen: false,
      newProjectName: "",
      openReloadRsrc: false,
      openAcceptTemplate: false,
      openNormalization: false,
      openOptimization: false,
      openComplex: false,
      itemid: 0,
      itemmcode: "",
      itemname: "",
      checkedA: false,
      loading: false,
    }
  }

  updateState() {
    this.setState({ loading: true })
    $.ajax({
      url: '/api/getProjects',
      dataType: 'json',
      contentType: 'application/json;charset=utf-8',
      method: 'GET',
      success: function (data) {
        this.setState({ projects: data, loading: false })
        this.forceUpdate()
        // console.log('Hello')
      }.bind(this),
    })
  }

  componentDidMount() {
    this.updateState()
  }

  render() {
    const { classes } = this.props
    return (
      <div className={classes.root}>

        <Table className={classes.table}>

          <TableHead>
            <TableRow >
              <TableCell >Код проекта</TableCell>
              <TableCell >Название проекта</TableCell>
              <TableCell >Степень проработки</TableCell>
              <TableCell >Статус</TableCell>
              <TableCell ></TableCell>
              <TableCell ></TableCell>
              <TableCell ></TableCell>
            </TableRow>
          </TableHead>

```

ROR') ?

```
</TableRow>
</TableHead>
<TableBody>
  {this.state.projects.map((item, index) =>
    <TableRow key={index} className={classes.item} hover>
      <TableCell>{item.code}</TableCell>
      <TableCell>{item.name}</TableCell>
      <TableCell>
        { /* {(item.stateName !== 'OK' && item.stateName !== 'NEW' && item.stateName !== 'ER-
ROR') ?
          (<img src={loader} alt="please wait" />) : ("))* */
          {item.stateComment}
        </TableCell>
        <TableCell>{(item.agree ? "Утвержден" : "Не утвержден")}</TableCell>
        <TableCell>
          <Tooltip title={"Редактировать"}>
            <Button
              color="secondary"
              style={{ textTransform: 'none' }}
              variant="contained"
              onClick={() => this.props.handleClick(item)}
              // disabled={(item.agree ? true : false)}
            >
              <Edit />
            </Button>
          </Tooltip>
        </TableCell>

        <TableCell>

          <SimpleMenu
            handleClickNormalization={this.handleClickNormalization}
            handleClickReloadRSRCOpen={this.handleClickReloadRSRCOpen}
            handleClickOptimization={this.handleClickOptimization}
            handleClickComplex={this.handleClickComplex}
            item={item}
          />
        </TableCell>

        <TableCell>
          <Tooltip title={"Утвердить шаблон"}>
            <div>
              <Button
                color="primary"
                style={{ textTransform: 'none' }}
                variant="contained"
                onClick={() => this.handleClickAcceptTemplate(item.id, item.code, item.name)}
                disabled={(item.agree ? true : false)}
              >
                <DoneOutline />
              </Button>
            </div>
          </Tooltip>
        </TableCell>

      </TableRow>
    )}
  </TableBody>
</Table>
```


Пример отображения таблицы и загрузки данных, серверная часть

```
func getProjects(db *sql.DB) func(w http.ResponseWriter, r *http.Request) {
    return func(w http.ResponseWriter, r *http.Request) {

        rows, err := db.Query("EXEC [dbo].[CyclePlanning_Template_Projects_Get]")
        if err != nil {
            errorWithJSON(w, "Database error", http.StatusInternalServerError)
            log.Println("Failed to run a query: ", err)
            return
        }
        defer rows.Close()
        var records []Record

        for rows.Next() {
            var record Record
            err := rows.Scan(
                &record.ID,
                &record.Code,
                &record.Name,
                &record.StateDate,
                &record.StateName,
                &record.StateComment,
                &record.Agree,
            )
            if err != nil {
                errorWithJSON(w, "Database error", http.StatusInternalServerError)
                log.Println("Failed to fetch result: ", err)
                return
            }
            records = append(records, record)
        }
        err = rows.Err()
        if err != nil {
            errorWithJSON(w, "Database error", http.StatusInternalServerError)
            log.Println("Failed to run a query: ", err)
            return
        }

        respBody, err := json.MarshalIndent(records, "", " ")
        if err != nil {
            errorWithJSON(w, "JSON Marshal Error", http.StatusInternalServerError)
            log.Println("Failed to marshal to the JSON format: ", err)
            return
        }
        responseWithJSON(w, respBody, http.StatusOK)
    }
}
```

Пример кнопки с диалогом подтверждения, клиентская часть:

```
class CreateTemplate extends React.Component {
    constructor(props) {
        super(props)
        this.state = {
            openCreateTemplate: false,
            projName: 'Новый шаблон',
            loading: false,
        }
    }

    createTemplate() {
        this.setState({ loading: true })
    }
}
```

```

$.ajax({
  url: '/api/createTemplate?name=' + this.state.projName,
  contentType: 'application/json;charset=utf-8',
  method: 'GET',
  success: function () {
    this.setState({ loading: false })
    this.forceUpdate()
  }.bind(this)
})
}

handleClickCreateTemplateOpen = () => {
  this.setState({ openCreateTemplate: true });
}

handleClickCreateTemplateClose = () => {
  this.setState({ openCreateTemplate: false })
}

handleClickCreateTemplateCloseDone = () => {
  this.setState({ openCreateTemplate: false })
  this.createTemplate()
}

handleChange = name => event => {
  this.setState({ [name]: event.target.value });
};

render() {
  const { classes } = this.props
  return (
    <div className={classes.root}>
      <Button style={{ textTransform: 'none' }} variant="contained" color="secondary" className={classes.button}
        onClick={this.handleClickCreateTemplateOpen}>
        Создать новый шаблон
      </Button>

      <Dialog
        open={this.state.openCreateTemplate}
        onClose={this.handleClickCreateTemplateClose}
        aria-labelledby="alert-dialog-title"
        aria-describedby="alert-dialog-description"
      >
        <DialogTitle id="alert-dialog-title">{"Создание шаблона проекта"}</DialogTitle>
        <DialogContent>
          <TextField
            required
            name={"projName"}
            margin="dense"
            onChange={this.handleChange('projName')}
            label={"Название шаблона"}
            fullWidth
            defaultValue={this.state.projName}
          />
        </DialogContent>

        <DialogActions>
          <Button onClick={this.handleClickCreateTemplateClose} color="primary">
            Отмена
          </Button>
          <Button onClick={this.handleClickCreateTemplateCloseDone} color="secondary" autoFocus>
            Создать
          </Button>
        </DialogActions>
      </Dialog>
    </div>
  )
}

```

```

</DialogActions>

</Dialog>
{(this.state.loading) && <img className={this.props.classes.fab} src={loader} alt="please wait" />}
</div>
)
}
}

```

Пример кнопки с диалогом подтверждения, серверная часть:

```

//CreateTemplate for MakeTemplate
func createTemplate(db *sql.DB) func(w http.ResponseWriter, r *http.Request) {
    return func(w http.ResponseWriter, r *http.Request) {

        name := r.FormValue("name")
        claims := r.Context().Value(jwtContextKey("decoded")).(jwt.MapClaims)
        username := claims["username"].(string)

        rows, err := db.Query("[dbo].[CyclePlanning_Template_Create]",
            sql.Named("Proj_Name", name),
            sql.Named("user", username))

        if err != nil {
            errorWithJSON(w, "Database error", http.StatusInternalServerError)
            log.Println("Failed to run a query: ", err)
            return
        }
        defer rows.Close()
        err = rows.Err()
        if err != nil {
            errorWithJSON(w, "Database error", http.StatusInternalServerError)
            log.Println("Failed to run a query: ", err)
            return
        }
    }
}

```